TOOLS FOR ACQUIRED SCENE ANALYSIS AND TRANSMISSION

BY

MAHSA KAMALI MOGHADDAM

DISSERTATION

Submitted in partial fulfillment of the requirements
for the degree of Doctor of Philosophy in Computer Science
in the Graduate College of the
University of Illinois at Urbana-Champaign, 2011

Urbana, Illinois

Doctoral Committee:

Professor John C. Hart, Chair
Professor Klara Nahrstedt
Professor Roy H. Campbell
Eyal Ofek, Ph.D., Microsoft Research
Matei Stroila, Ph.D., Navteq Research

ABSTRACT

Recent advancement in research and technology on virtual environments has significantly escalated users' expectations on this matter. This trend is easily witnessed, for example, by websites like twitter or facebook. Moreover, searching for 3D navigation in any search engine can provide an increasing number of instances where the full 3D models of cities are available to users for a virtual walk through. From research viewpoint, design of such virtual environments needs to address two key concepts of “user interface” and “data manipulation”.

A wide spectrum of user interfaces has entered into the industry of virtual navigation. These range from haptic devices such as joysticks, Wii, and touch screens to the more sophisticated options such as Microsoft Kinect. Obviously, the potential of these devices needs to be carefully evaluated in each application to strike a balance between efficacy and complexity. In this thesis, we have investigated a few case studies on quite different tasks to justify the universality of this issue. The tasks vary from simulated circuit design to GPS navigation. Of particular interest is the task of choreography software, where the interface evolved from dancers' personal experience into an optimal combination of preset virtual scenes and real time control of the scene by a Wii device.

Data manipulation methods vary depending on whether 2D or 3D data is being used. The choice between 2D versus 3D depends on the available bandwidth and processing power. For example, when working with navigation maps on mobile devices, one cannot easily transmit huge 3D data. Hence, researchers tend to either use graphical models or 2D images. On the other hand, when high bandwidth and processing power is available, it is often better to use full 3D capabilities. Regardless of the choice of 2D or 3D sensor, there are certain limitations for each one. For example, 2D sensors often suffer from limited field of view which is a major strain when user wants to get a sense of the entire scene. On the other hand, while 3D sensors can potentially provide 360 degrees view of the scene, their captured data is too noisy and sparse to be useful for any automatic structure detection. In this thesis, we address both 2D and 3D data and develop novel techniques to remedy the aforementioned problems within specific applications. The ultimate results of these works provide robust enhancement and detection in 2D and 3D data. We briefly overview the 2D and 3D tasks here and leave the details to their associated chapter in this document.

We study two 2D tasks focusing on issues of robust detection and panoramas creation respectively. The first task, which we named Meth Morph, analyzes 2D portrait images of human face and morphs them into a face as if the subject is drug (Meth) addicted. The developed technique requires only a single input image and performs the rest of the task in a fully automatic fashion. The robustness of the systems is perhaps best justified by referring to the successful live demo given to random visitors.

This thesis also investigates overcoming the limited field of view in 2D cameras by available panorama creation methods. A very beneficial application for panoramas is being able to view a long street in one single view. For example, when working with mobile devices, one needs a highly efficient way to transmit and display a full length street. This was made possible by a project called Street Slide at Microsoft. Nevertheless, clutter over the pace of stitched images (e.g. telephone wires) can seriously damage the performance of street panoramas. To improve the robustness of panorama creating, in this thesis, we develop a fully automatic detection and removal of wire-like clutters in the images.

Finally, we investigate robustness issues in 3D data processing and structure detection. The first challenge with processing 3D data is their huge size compared to 2D. This makes their processing computationally expensive; an issue that is in conflict with their real-time use as in interactive environments. For this purpose, we develop an efficient system based on an OCtree spatial data structure allowing real-time rendering and manipulation of a 3D point cloud. The other challenge in processing 3D data is the issues of noise and outliers. In this thesis, we develop a novel method for robust classification of curvilinear and surface like structures within a point cloud.

All the mentioned pieces help improve the efficiency and quality of data processing, analysis and even transmission. The latter is of special demand by nowadays technology of mobile devices.

To the best of the best whom I love and owe all my success to

My life companion

Hossein Mobahi

&

My mom and dad

Mahrokh Soltani and Kiumars Kamali

ACKNOWLEDGEMENTS

This PhD could not have been possible without guidance, support and deep knowledge of my thesis adviser Prof. John C. Hart. He created a lively group in which all members willed to help each other with joy and excitement. I cannot say enough about his great support and involving me in useful and joyful opportunities such as SIGGRAPH, MIDGRAPH, 3DPVT etc. I always bragged to my friends on him being a great reference point wherever I had issues in my research or other university matters. I deeply thank John for being such a great and supportive adviser.

Since last year that I first met Dr. Eyal Ofek, I was amazed by the depth of his knowledge and how fun he made every complicated problem to solve. While I was an intern in Microsoft, we named him “idea gun” for quickly coming up with lots of great ideas as soon as you give him a problem. I have been so lucky to have his support and guidance throughout the past year and I am looking forward to collaborating with his group after joining Microsoft.

Having the opportunity to be an intern at Navteq, I was very lucky to work under supervision of Dr. Matei Stroila. I was impressed with his great enthusiasm and terrific mathematical knowledge. He was very supportive and always available for help whenever I needed his advice on my research.

I had a productive and fun time working with Prof. Klara Nahrstedt's team on the TEEVE project. She is a great supervisor who was able to manage more than eleven students at the same time. Working in her team even as short as a few months was so joyful and productive that resulted in four of my ACM publications! In simple words she is just a super woman with true dedication for her students.

When I first arrived with my husband to Urbana-Champaign, the first UIUC professor and the kindest one I have ever met, Prof. Silvian Ray (may he rest in peace) saw me behind my husband and asked about my background. When he talked to me a bit, he said "you will be a great student here, why don’t you apply?!" His simple sentence has had a huge impact in my life and because of it now I am here, about to finish my PhD.

Soon after being encouraged by Prof. Ray, I searched for projects in Siebel Center that were relevant to my background and got familiar with the fantastic Cultural Arts team of Prof. Guy Garnett and Prof. Roy Campbell. I was privileged to soon become a team leader in their class

project and improve my self-confidence. Later, Prof. Campbell introduced me to John’s fantastic graphics team! I deeply thank his great support.

I would also have to mention the great support from Prof. Harandi who was always there to guide me through department's administrative issues and correcting my path whenever needed. He is another one of the great gemstones in Siebel center! I similarly feel the urge to thank Prof. Geneva Belford, Barb Cicone, Laural Heriot and Mary Beth Kelly, and Siebel's TSG group, who always and instantly took care of my problems and questions. I also need to thank Prof. Thomas Anastasio, Prof. Karrie Karahalios and Dr. Eric Schaffer for their invaluable support and feedbacks at different times. They helped me improve the breadth of my knowledge across quiet different research areas. Moreover having the opportunity to participate in the fun and productive Vision Lunches and guidance from Prof. David Forsyth and Prof. Derek Hoiem I could enrich my computer vision knowledge. I also need to thank Prof. Yi Ma and his group for helping me with my questions and always treating me as a member of their group. Most importantly, Yi provided extremely helpful advices to me and my husband throughout the very hard time we were having due graduate school hardships.

I had a great time with and learned from my wonderful lab mates Keenan Crane, Yuntao Jia (and his wife, Jennifer), Victor Lu, Kevin Karsch, Jared Hobroek, Rajesh Bhasin, Apeksha Godial, Forrest Iandola, and Shadi Ashnai. Having such wonderful friends around me every day doubled my energy for doing research. I also like to mention my wonderful friends in the "women in computer science" group: Erin Chambers, Anna Yershova, Kamilah Taylor, Nana Arizumi, Sruti Bandha, Sonia Harris, Katrina Gossman and Adair Liu and all wonderful WCS officers. With being acquainted with them I always felt loved and cared for and had so many fun days. I specially owe great thanks to Homa Alemzadeh and Farzad Farnoud for being my study pals and always there with me to study. It would have not been this much fun staying long long nights till morning in Siebel without them. I wish them successes in their academic path.

I also had a wonderful moral support group in Champaign including, Shahpar Taheripour, Farzad Tahripour, Farnaz Soheili, Nayer Porkhosrow, with whom I found Champaign to be the happiest place on earth. Those and close friends Mojgan Jamali, Afarin Jahangiri, Hamid Jahani, Arezoo Khodayari, Afsaneh Shirazi, Leila Fuladi, Bita Vaezian and Mani Golparvar were always there when I felt alone and needed company to calm and relax. I also need to mention the great friendship and fun times I had here with Tony and Lin Bergstrom, Jacob and Jesse Schorloder, Zihan Zhou, Omid Fatemiyeh, Parya Moinzadeh, Shankar Rao, Andy McGovern, John and Jenny

Heaton, Amir Nayeri, Dan Schriber (may he rest in peace), Dmitry Yershov, Riccardo Cerepaldi, Mirco Montanary, Joshua Hailpern, Marina Danilevsky, Alexander and Daria Sorokin, Yoav and Alina Sharon.

I couldn't have been here without the unconditional love I received from my mom Marhrokh Soltani and dad Kiumars Kamali. I owe every little success in my life to them. I cannot mention enough about my mom's great sacrifices such as scanning hundreds of pages of my book which couldn't be posted from Iran in time to me. She also went through all the terrible travel difficulties despite her being physically not in good condition to be here during my defense. My dad also passed extremely difficult days when he tried to get a visa and visit me here, but he didn't give up till he could finally get here. Needless to mention, these are very small portions of the huge sacrifices they have made for me. I love them so much because of their great support and unconditional love throughout my life. I couldn't be more proud and happy having such wonderful and loving sisters Parisa and Melika who have been my great companions and friends. I cannot mention enough about the happy days and memories we have shared together. I want all the best for them and wish all their dreams to come true.

I have to ultimately thank my sweetest grandparents MamaMina (may she rest in peace and remain always in my heart), Baba Bozorgh and Maman Joon for always believing in me and loving me so much. I need to mention my wonderful parents in law Maghbouleh Haddad and Khalil Mobahi, sisters in law Haleh and Hedyeh, and brother in law Shahram, who have been a great moral support to me. Also I always brag about having my aunt Mojgan and uncle Shahrokh as my very supportive family members in United States and thank them for not letting me feel alone here. My cousins in law Bita Shakoori, Payam and Parisa Pooyan have also made a huge difference in my adventurous path in the United States. I am so lucky to have them as my wonderful friends.

Lastly but most importantly, I could not enjoy my academic and personal life without my great and most caring life companion Hossein Mobahi. Hossein opened my eyes to a whole new world of life adventures and success which would have been impossible have I not been bonded with him. Despite him having the same hardships I was facing as a student, he has always been there to support me even when he was himself in huge stress. I have shared the best days of my life with him and cannot be any happier in the journey we are experiencing together. I have to deeply thank him for loving me so much and making many unconditional sacrifices I could not even imagine. I could not even think of being here and succeeding in UIUC without having him beside me.

Table of Contents

CHAPTER 1
Introduction

In the past decade, expectations from user interface and simulation environments have significantly raised. With the growing market of 3D televisions and 3D scanning technology, research about enhancing 2D or 3D navigation environments has become of special interest. The two fundamental pieces of this subject include user interface and data handling. Obviously, in order to take full advantage of each of the mentioned pieces, the other pieces must be functional as well. In this manuscript, we present our findings and novel improvements in each of the mentioned aspects. Finally we also suggest some directions for further exploration and enhancements.

In this chapter, we briefly discuss technologies related to user interface development and their effectiveness. Subsequently, we deliberate our findings on data handling in such systems. Finally, we explain our novel expansions over three different data handling challenges on 2D and 3D scanned scenes for the purpose of better transmission and understanding.

Chapter two discusses our previous findings and results in the user interface enhancement. Chapter three will talk about the main achievements in this thesis over the data handling problems we face. Finally on Chapter four we explain our conclusion and future research expansion possibilities.

1.1 User Interface

There exists a wide range of new interface hardware devices. Three great examples from this spectrum are Nintendo Wii [1] and Multi-Touch Screens (such as iPhone or iPad [2]) and Kinnect technology from Microsoft. With the latter, users don't need to hold any device in their hands in order to interact with a virtual environment. Due to such a broad spectrum of technologies, the user interface software designers need to carefully study the effectiveness of the chosen device for their program. This task becomes very challenging when dealing with 3D environments controlled by 2D based interfaces such as multi-touch surfaces or mice. Thanks to the new haptic device technology, this challenge can be partially alleviated for surfing 3D environments [3].

After choosing a proper hardware for an environment, next step is to fully harness all available hardware interface capabilities. However, exploiting a larger set of interfaces should not complicate its use. For example, consider a mission to be controlled mainly by user's body movement. Here, Wii is a very intuitive choice for the user to direct the task. However, when the task involves a large action vocabulary, it becomes impractical to encode each action by hand movements; it gets difficult to learn which sequence of hand movements leads to which action. A possible solution is to split the control between the haptic device and a keyboard. The keyboard can switch among smaller and more intuitive subsets of Wii actions. This must be done carefully, so that the actions likely to occur together are grouped to the same Wii action subset. This helps the user to use the keyboard as infrequent as possible, yet enjoy an intuitive control mechanism via Wii. Therefore, interface software design is highly oriented around user studies and exploring what can be the most “intuitive” and “convenient” way to control an environment using a particular set of control devices [4], [5].

Although there exists a broad set of choices for 3D haptic devices, controlling complex virtual environments requires additional support from a graphical user interface (GUI). Going back to the example in the previous paragraph, while by using a Wii remote, the user might be able to move objects or the environment, but not all the capacity of a complex system could be efficiently controlled by a Wii device. Switching among various operation modes requires an intuitive graphical interface so that the user can tune it with minimal amount of training. An example of this is a hierarchical menu system, where the menus are organized based on their correlation. Hence, the intuitive control of the environment is one reason for needing GUI in such systems. Another reason for having a GUI in such environments is to let the user experience a change in the status of the environment while engaged in an interaction with the system [4], [6]. The status of the environment could change simply after a fixed amount of time is passed, or as a response to some stimulus coming from the user. The former could be the case when certain virtual objects or environments must appear at certain times throughout a virtual dance setup. The latter, for example, is used in GPS systems to update the location of the vehicle along the map, based on satellite signals.

An ideal GUI design for a complex environment must integrate all the mentioned components (hardware capabilities, intuitive interface, efficient menu hierarchy, and change in the status of the environment). A recent approach for achieving this goal is creating several scratch models of the interface and tuning each model by studying it on real users. This technique is easily applicable to

internet users where several sample websites are generated and each is assigned to a certain IP group (i.e. users). Then, the website model which gets the highest user hits is chosen as the final interface [7]. In CHAPTER 2, we explain different experiences that we had with user interface design and enhancement and present the outcome of the studies based on the created and enhanced platforms.

1.2 Data Manipulation

As mentioned earlier, another important success metric for having a decent 3D virtual navigation environment is satisfying users' expectations on visual effects while economically utilizing bandwidth and time. For instance, GPS review websites run several forums to discover to how much quality is required for 3D rendering so that users definitely prefer it over 2D representation [8]. In fact, currently one of the main competitions in navigation map industry is generating high quality 3D maps for use in GPS systems [9]. As an example, on November 18th of 2010, Navteq bought PixelActive Company[1] in order to get their help on improving 3D city models. Achieving high quality 3D graphics, even in presence of most modern laser scanning and stereo imaging equipment, has remained a challenge and the competition in this industry is going on.

An alternative application of 3D reconstruction is Tele-Immersive environment for virtual collaboration in 3D[2]. A couple (out of many) applications in such systems are teaching dance movements remotely, guiding patients for correct therapy related actions by doctors remotely and gaming. While modeling a realistic world is not the top priority in these systems, being able to immerse characters realistically and efficiently in augmented reality is crucial. Tele-Immersive environments involve processing and transmitting/receiving a huge amount of 3D data. Therefore, 3D data processing must satisfy real-time constraints in such systems [4].

[1] http://www.navigadget.com/index.php/2010/11/18/navteq-buys-pixelactive-for-better-3d-maps
[2] http://cairo.cs.uiuc.edu/teleimmersion/

An important influence on 3D data is to reduce the size of it. This process keeps the desired details intact, but removes the redundant information. The reason for accepting such immense data reduction techniques is because, small devices such as GPS systems or PDAs are not able to store and render huge amount of point cloud data in them. However, rendering 2D images or a simplified model is an achievable goal for such devices. Two examples of such techniques are imitating 3D vision by passing 2D image warps [10] and model based rendering [11].

One of the ways for being able to reconstruct the world is extracting needed and meaningful information from data in order to create a model from it. The challenge regarding extraction of data differs between 2D and 3D data. This is because, while in 3D the information content has one more dimension comparing to 2D, most regular scanners are not able to generate noiseless and uncorrupted data. Therefore, typically it is needed to fuse multiple information sources in order to suppress the noise and recovering the corrupted or missing data [12]. Note that these steps are not needed for 2D image data. Therefore, some items are very easy to identify in 3D but difficult in 2D and vice versa. We can also categorize feature extraction into local and global categories, where in local feature extraction, only a small neighborhood feature set is considered and in global feature extraction, all the data affect each other in terms of feature elements. For example, if one item is classified locally as class A, while all its neighbors belong to class B, then, we can assume this item most probably also belongs to class B. We have experienced feature extraction from both types of data and will explain them in CHAPTER 5.

The other approach for reconstructing data is looking at it from compression point of view. This typically accounts for removing parts of the data which do not affect users' perception that much. One example of this type of reconstruction in 3D is reducing the number of points based on neighborhood density. In 2D this can be by removing, blurring unimportant parts of the scene such as background [13]. Another widely used example is background subtraction in 2D images. The compression plays a great role in significantly reducing next steps calculation times and incorrect bias of algorithms to redundant data.

The method of warping is suitable for systems where limited bandwidth is the highest concern. Here, we transmit the 3D structure in 2D pieces of information [10]. The client then warps them according to the desired view. This technique requires a wise choice over sets of images to be sent for clients. In section 3.2 we will discuss our recent research regarding this.

CHAPTER 2
User Interface

During the past few years, I got involved in multiple projects related to navigation technology and research. This chapter provides a brief summary of our research results obtained so far. We categorize the topics in a similar way to the introduction. First, we explain our experiences over user interface design and enhancements. Next, we address data analysis both in 2D and 3D. Finally, we discuss data reconstruction. Here, past research on three separate user interface design projects are explained. To improve readability, each one is explained separately.

2.1 Haptic Device Controlled 3D Environment

In this project the goal was creating haptic device to instruct a software program used for designing semiconductor boards. Users could burn certain parts of the board (based on the circuit design model) by moving a simulated robotic pin. The user achieves/experiences this using simple haptic movements and feedback. At the beginning, the user moves the pin to a desired location on the board and then presses the pin to burn the surface. The haptic device generates both sound and bumping feedback to alert user about potential problems (such as burning a place for too long or an incorrectly burned spot). This platform was very useful for high school kids interested in learning semiconductors design in a controlled and simulated environment. Particularly, the bumping feedback turned out to be more effective than the audio feedback. Hence, this type of haptic interface can be useful for inducing similar feedback, such as collisions in Tele-Immersive environment or obstacle (alerts) for remote vehicle control. The mentioned work was published as a poster in national center for learning and teaching in Nano scale Science and Engineering[3].

[3] http://www.nclt.us/grg/34057.pdf

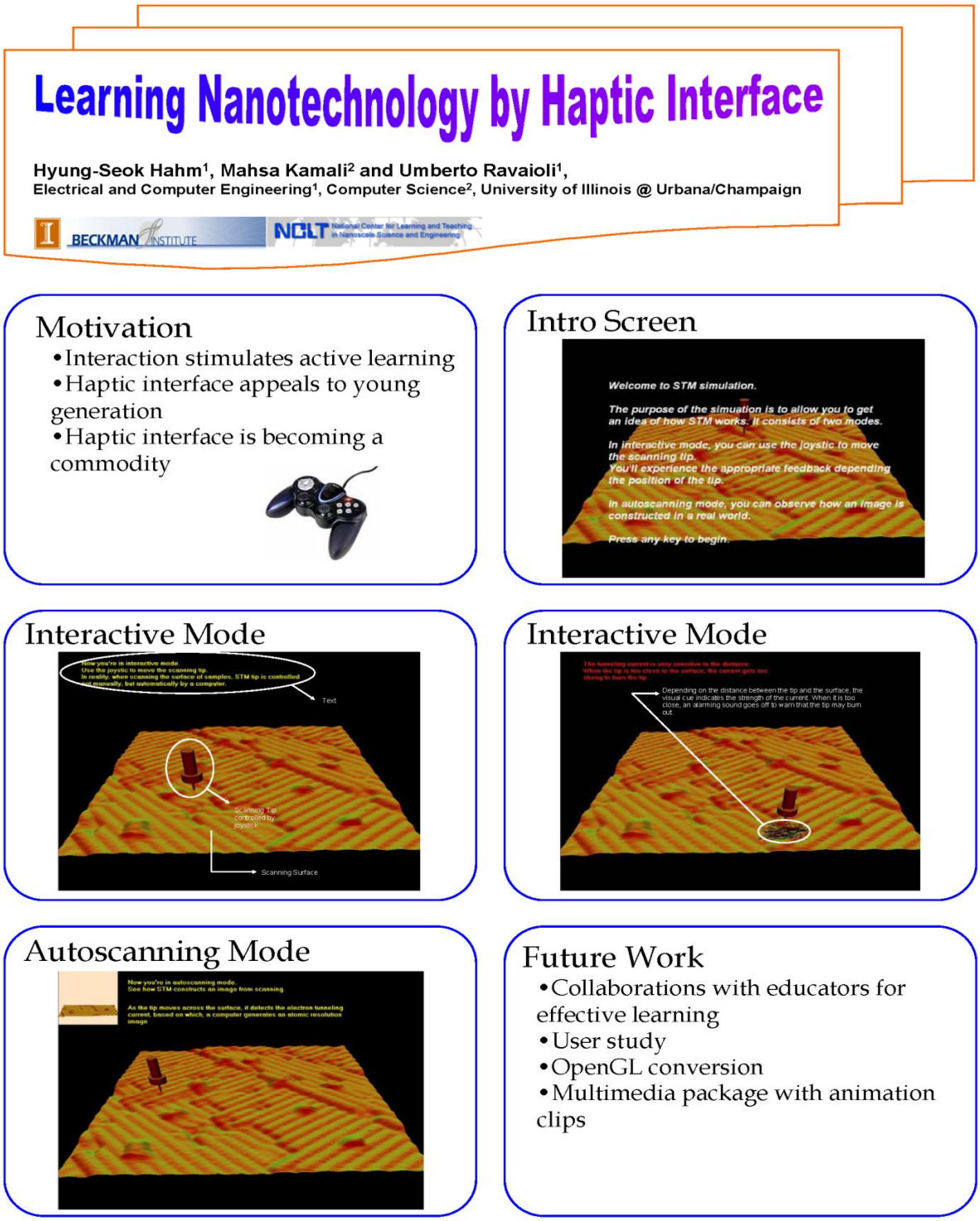


Figure 2-1 Haptic Interface for Semiconductor Board Design

2.2 3D Virtual Dance Environment Interface Design [4 , 6]

In this project, we explored two ways of controlling a graphical environment, namely by a haptic device (Nintendo Wii) and a graphical user interface. The combination provided a number of options for the users to design their virtual environment. We studied how the design of such interfaces could enhance a virtual dance environment. We also discovered which options were more appreciated by the dancers in this study. We understood that the expectation of an average art user from a virtual environment is much higher than what the current technology can offer. However, users were always enthusiastic to design the environment in their own way. Figure 2-2 shows the layout of this interface and environment.

Figure 2-2: TEEVE graphical user interface and Wii control

We designed the graphical user interface by discussing it with the dance teacher for the class. She wanted the dancer to be able to view graphical objects and throw them into a timeline (like a video editing platform). Figure 2-2 *left*, shows two of the objects that users dragged and dropped into the timeline (first one, between minutes 0 till 29, is a terrain and the second one, between minutes 3 till 16, is water). In general, with this platform they were able to design a time triggered object appearance in the scene and then control the objects behavior with the Nintendo Wii device. The dancers found playing with Wii more fascinating than designing a timeline. Nevertheless, they found the existence of such timeline interface helpful.

In this project we figured the essential need of considering the users expectations and skills while designing an interface. Such interface comes very handy, when working with a collaborative environment, such as gaming, or dancing remotely. However, with a navigation device, automation is always more essential than real-time control of the system by the users. Therefore, using a haptic device such as Wii is not suitable for a GPS device unless someone would like to control navigating through a map model like a game or remote vehicle control. At that stage, a haptic device like a wheel joystick or Wii would be more suitable than mostly used touch interface of such a device.

2.3 Map Navigation on Touch Interfaces

A new but widely available set of interfaces are touch capable PDA devices such as iPhone or iPad. In fact, Google and Microsoft have dedicated special groups focusing on such interfaces for their navigation map websites. First major concern when utilizing these devices is ensuring that the action gestures are “easy” and “intuitive” for a general user. The other significant issue here is that not all of these devices work so perfectly as other types of interfaces. Finally, not all of them are multi-touch capable.

Figure 2-3: Multi-Touch and Single Touch Map Control Gestures

For example, the two multi-touch gestures shown in Figure 2-3 are not available on single touch devices. Therefore, when designing software for web based navigation, interface designers should always consider an alternative action for the less capable platforms. This is particularly important for vital actions which “must” function in order to navigate around. For example, performing a zoom in/out should not be restricted only to pinching by two fingers. Instead, it should also support a replacement gesture such as double tapping. Same goes with rotation, for which the designer must place graphical tap-able buttons for rotation.

2.3.1 Problems

- Each touch capable client must be able to use all features in Bing maps with intuitive and error prone touch events without need to use any other user interface (ex. Keyboard or mouse).
- With current vastly growing market on touch screen devices such as PDA’s, Monitors and Tablets, one of the key elements of success in becoming consumers number one choice is to provide them understandable and smooth map control which can easily get benefits from current touch device capabilities. (Market analysis provided in further sections)
- Currently, a few of the platform events have no touch event dedicated for them (ex. BE Rotation). Moreover, some have problematic touch support with unexpected outcome (ex. Hold

press for right click causing flicker). This spec focuses on finding where the mentioned issues are and suggesting brief solutions.

2.3.2 Goals

- Natural and Intuitive Interface
- Creating Direct and Engaging Experience for all possible features
- Ease of usage
- Enable Confidence in Users (Eliminate Surprising behaviors or Errors)

The list of items related to each feature is shown below in priority order. Also you can find more details on Appendix A.

Task	Notes
Top Bar and its Properties Page must be touch functional.	Text Boxes, Hyper Links, Radio Buttons, Check Boxes, Buttons
Basic Navigation Must be fully touch functional	Zooming, Panning, Rotating, Buttons
Basic Search Must be touch functional	search text box, left pane, POIs, Nearby Listings, Hyperlinks, Scores
Page Interaction and Manipulation Must be Touch Functional	Zooming, Scrolling, Full Screen, Text Selection for Copy/Paste
All actions must be functional without need for hovering except tooltips.	Including: POI info, Push Pins, Search List
Routing Must be fully touch functional	Scroll down menu for (walk/drive/ transit), Text Boxes, Left Pane Scrolling and Step Selection, Start/End point and road dragging
Sharing Must be Touch Functional	Text box, Navigation, Radio Button, Check box, Drop Down Menu.
My Places Menus Must be Fully Touch Functional	Drawings, Tabs, Hyperlinks, Push Pin Drops, Drop Down Menus
Printing Must be Touch Functional	Text Box, Page Zoom and Scrolling
Mini-Map Should Be touch Functional	Opening, closing, panning
Everything Should be zoom-able to resolve most Fat Finger issues	Hyper Link Text, Buttons, Menus,
Full Screen and Maximize to Left Pane Should be Touch Friendly	Enable/Disable Button

CHAPTER 3
Data Analysis and Enhancement

The acquired data from interface devices often require analysis or manipulation to produce an appropriate output. In this section, we explain some of the related works on analyzing or processing visual data.

3.1 Self-Aiming Camera

The self-aiming camera consisted of three major acquisition devices: a low resolution infra-red, a wide angle and a close up camera. This system was a surveillance system with the ability of autonomously detecting and tracking interesting targets [14]. A regular surveillance system has to use a single angle focus for a scene. However, since usually these cameras need to cover a big scene, they won't save high details for small objects in them. Figure 3-1 the middle picture shows one example of such wide angle view. Hence, if something happens in one part of the scene, usually, details are hard to get extracted from the distance and low resolution information. With the use of self-aiming camera, the close up view of moving characters in the scene could be captured because it was able to track and focus on the object of interest.

Here we briefly explain some technical details. With the aid of infra-red camera, the warm points in the scene (usually humans) were easily extracted. In addition, pixels belonging to a moving region/object were separated from still ones. The close up camera would focus on the pixels with highest heat and motion value. However, in order to avoid locking in one single view in consecutive frames, a negating doughnut was enforced around the chosen pixels. This reduced the chance of the same pixels to get chosen in the next frame. As a result, the zoomed view kept changing over various moving and heated areas in the scene.

A shortcoming of this system was being the fact that still; the system was expensive to build. In fact, even the cheapest infra-red cameras back then were around thousand dollars. Another problem of the system was the noise created by the pan-tilt unit for the zoom camera.

Figure 3-1: SAC (Self Aiming Camera), Middle picture shows the wide angle view and right image shows the close up from moving person in the scene.

By working on this project, I researched on detecting point of interest and generating saccades to ensure covering the entire scene. Both of these points are very useful in Tele-Immersive environments as well as in gaming for automated tracking. The system was installed for a while in Piccadilly store for tracking customer activities instead of their regular surveillance camera.

3.2 3D Video Image Efficient Transmission [10]

While feature extraction from data can lead to creation of simulated models from the data it shall not be the best option in some systems. This is due to experiencing significant information reduction which might not be desirable for viewers. Moreover, the expensive cost for calculating the output of global feature extraction algorithms makes them even more undesirable. As a result I have also been involved in systems where fast capture and enhanced visualization was sufficient answer to the needs of the system.

Streaming and rendering dynamic 3D data in real-time requires tremendous network bandwidth and computing resources. We created a remote rendering model to better study different remote rendering designs and define 3D video rendering as an optimization problem. Moreover, we designed a 3D video remote rendering system that significantly reduced the delay while maintaining high rendering quality. We also proposed a reference viewpoint prediction algorithm with super sampling support that requires much less computation resources but provides better performance than the search-based algorithms proposed.

Such a system is very useful for mobile devices that have very limited rendering and slow processing capabilities. In fact, rendering simple 2D images which are generated from warping the input images to the desired view is a great alternative for viewing data in such systems. Figure 3-2 shows the design and configuration of our system.

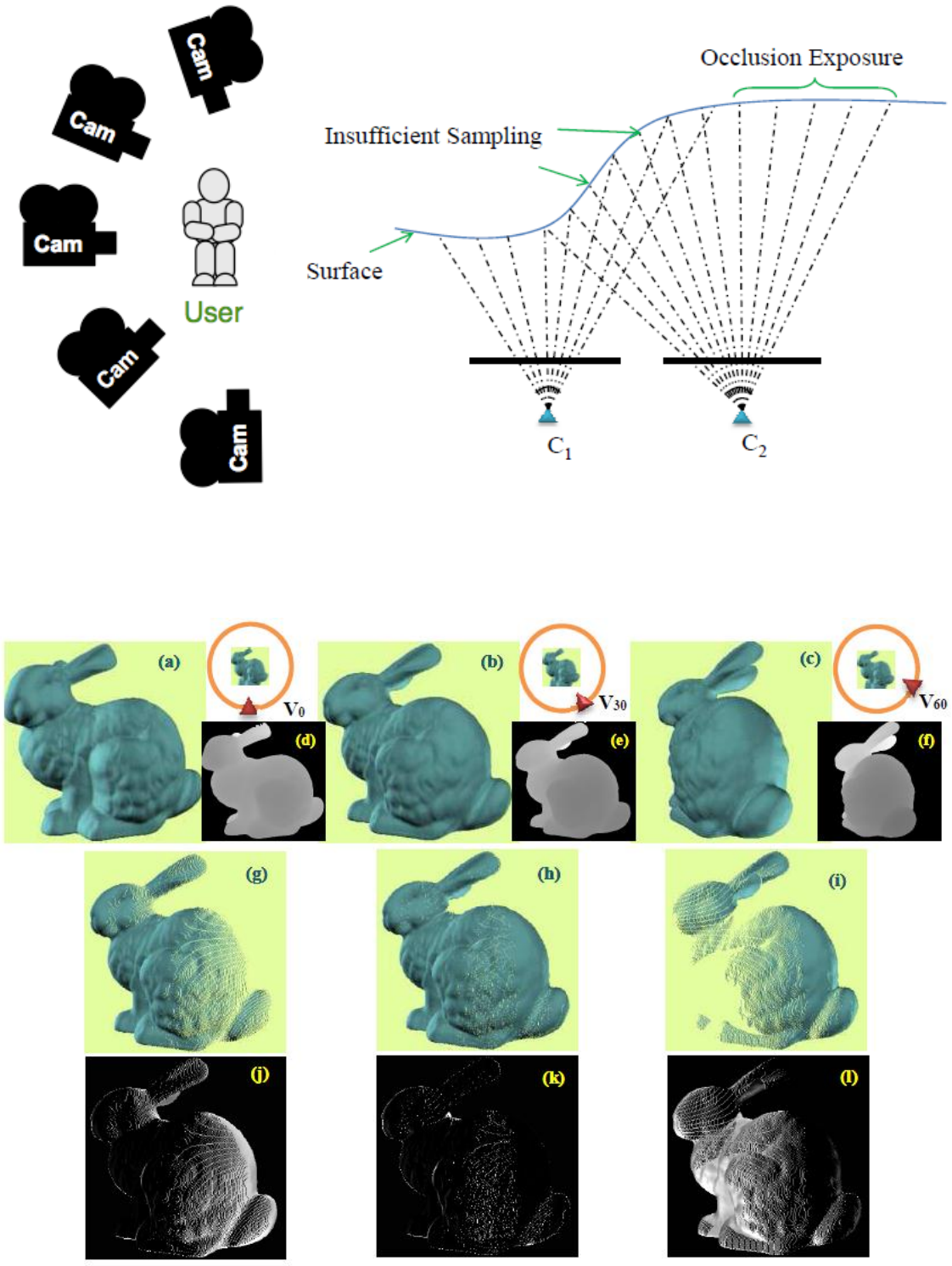


Figure 3-2: Design of the High Quality-Low Delay Remote Rendering System

3.3 3D Data Manipulation (Using OcTrees and 3D Graphics)

We needed an efficient data structure for keeping the 3D information as the structure differs from simple grid based 2D images. We chose and developed our own version of OcTrees. Our development uses nearest neighbor, k-nearest neighbor, and epsilon ball search algorithms. There are several research projects that solely focus on such hierarchical data structures, including a research done for parallel data processing by UPCRC in UIUC. However, OcTree was more than enough for our purpose of editing and a chunk of the data.

Our final platform allows the user to trim/cut/modify parts of the input point cloud data interactively by a mouse. The platform uses ray intersection technique to find the closest point in the OcTree to the pointing ray from the mouse. The user could use the mouse to create a sphere with at a desired location and radius. The sphere selects the portion of the point cloud subject to further editing. Such platform came especially handy when we wanted to label data for training the system how to do point classification. This task will be addressed later in this section for feature extraction in greater details.

One of the easiest ways to analyze 3D data is examining the local geometry information. For example, this could check if a region in the point cloud looks like a cylinder, sphere or disk using local principal component analysis [15]. The local geometry can be combined with supervised learning techniques to achieve an automatic labeling of the regions within a point cloud. For example, Figure 3-3 shows one of our results on automatic labeling of a tree. The image on the left shows neighborhood points chosen for PCA (the colored lines on it show the local principal axes). The image on the right displays the classification outcome, where the leaves and branches are labeled by green and brown colors respectively. Observe that the local geometry can help automatic labeling of these entities. For example, in branches, one principal axis is significantly larger than the other two.

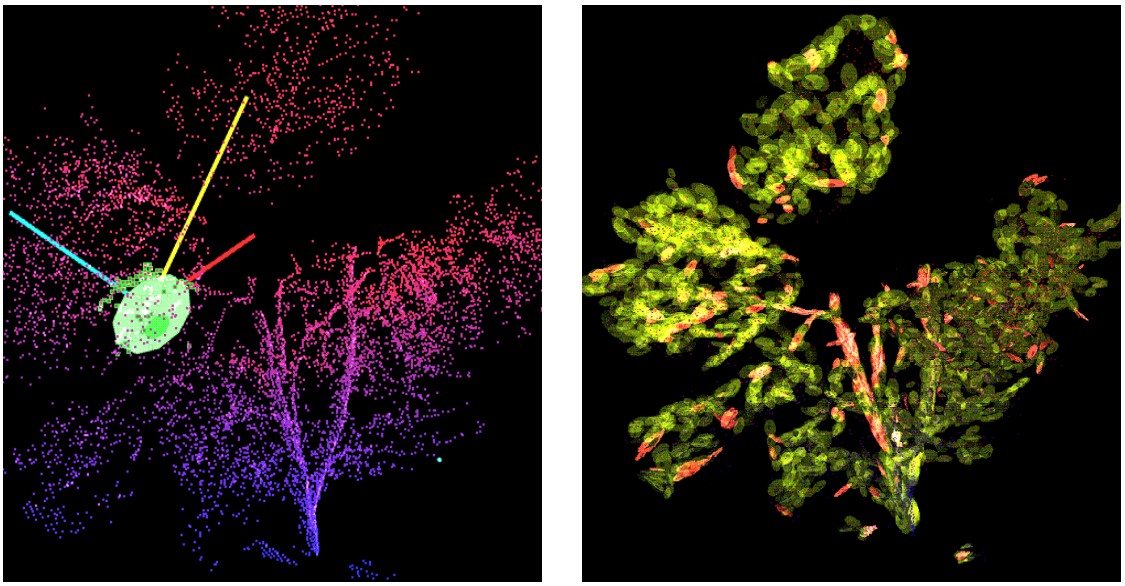


Figure 3-3: PCA based local feature finding and point classification (Leaf vs. Branch)

As the figure reveals, local features information is not enough for a good labeling. For example, due to the absence of knowledge that branches should be connected structures, they are sometimes isolated in the figure. As a result, we needed to include global constraints to glue local results in a coherent way.

We studied two techniques for imposing global constraints on the local features of 3D point data. The adapted Hough transform for finding (thick) tree branches or trunks. We also adapted Markov Random Fields for identifying vegetation areas. Both techniques performed well on clean and noiseless laser scan data. However, they had severe issues when exposed to Velodyne LIDAR[4] data that we were working with. Developing an appropriate noise model for laser data is still an open problem to us.

[4] http://www.velodyne.com/lidar/lidar.aspx

3.4 3D Sign Extraction

During summer of 2008 I was an intern at Navteq company. My research there was about extracting signs from 3D LIDAR data based on retro reflectivity. The work resulted in a patent. Since the traffic signs are retro reflective, the intensity level of the associated pixels in the capture device is significantly higher than other points. Figure 3-4 (*left*) shows these points in green. Hence, we are left with a small set of remaining points to find the sign.

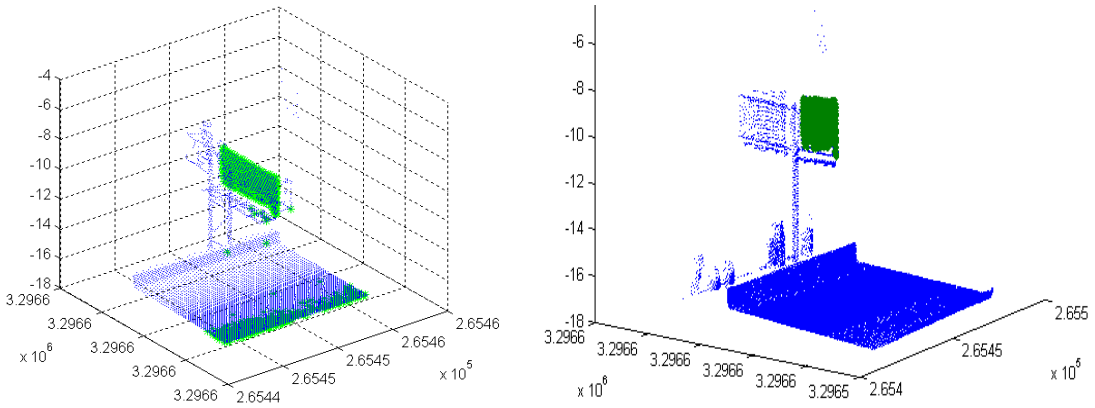


Figure 3-4: Sign Extraction from 3D LIDAR Data

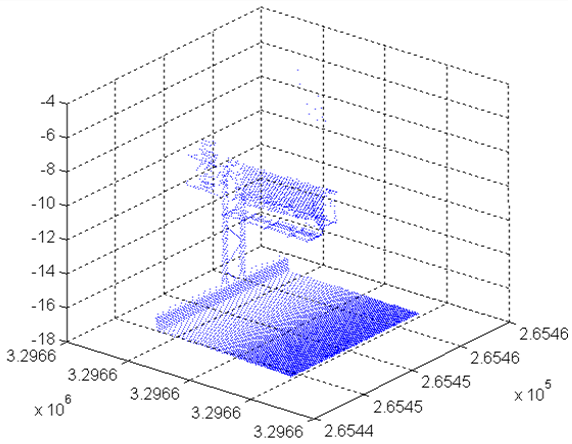
First, we locally extract the points which have a neighborhood plane normal perpendicular to the ground. Then RANSAC algorithm refines the initial guess about the plane and eventually returns the plane which the sign belongs to as shown in the Figure. Researchers at Navteq continued this project [16].

This type of characteristic based data extraction made us learn that extracting data from laser scans can be much easier when having extra information such as intensity or color considered. For example, while many objects in a scene can be hard to distinguish in 3D, such as different faces of a building, by adding the texture and color information these parts can be easily identified.

3.4.1 Method Overview

The input data is a 3D point cloud, where points are described by the 3D coordinates, (X, Y, Z), and reflectivity, I, (see an example in Figure 3-5), and an axis in this point cloud that corresponds to the driving direction. The data we use for exemplification was captured with the LYNX mobile mapper from Optech (http://www.optech.ca/pdf/LynxDataSheet.pdf).

Since signs are retro-reflective, we extract the highly reflective points from the scene. We compute a normal vector for each of these points by fitting a local plane. Then, we filter out the points that don't look towards the driving direction. We create clusters with the remaining points using a region growing algorithm described in [17]. We filter out clusters that have few points (less than 20 points) as noise. For each cluster we use region growing on the original point cloud (due to weathering, there are parts of a sign's surface that are not highly reflective), and merge overlapping grown regions. The last step is to fit planar geometry to these regions. Currently we are only fitting rectangles and are planning to support other shapes of signs to the method. Our results show a reasonable accuracy, not yet quantified as more data is needed. If images registered with the point cloud are available, they can be used to eliminate false positives and to reconstruct the sign as described in [18]. We don't expect to have many false positives since heuristics like size, location, and height above the ground can be used.



X	Y	Z	I
265442.970396041110	3296569.005907060100	-17.220000088215	199
265443.140439986430	3296569.005907060100	-17.223999798298	255
265444.050474166110	3296569.005907060100	-17.221999943256	252
265444.580381392680	3296569.005907060100	-17.227999985218	219
265444.750425337990	3296569.005907060100	-17.226000130177	186
265444.930479048930	3296569.005907060100	-17.227000057697	255
265445.110410689550	3296569.005907060100	-17.227000057697	252
265446.890439986430	3296569.005907060100	-17.215999901295	206
265447.070371627050	3296569.005907060100	-17.215999901295	224
265447.250425337990	3296569.005907060100	-17.223999798298	218
265447.430479048930	3296569.005907060100	-17.215999901295	253
265447.600400923930	3296569.005907060100	-17.222999870777	199
265447.780454634860	3296569.005907060100	-17.234000027180	255

Figure 3-5 Sample Input Data

Since signs are retro-reflective we filter out the points that don't have high intensity, as shown in Figure 3-6. Working with several data sets, currently the best threshold is 220 (the intensity value is between 0 and 255). We notice that some parts of the street surface can have also high intensity value. Therefore, it is impossible to use only the intensity as a measure for finding points corresponding to signs.

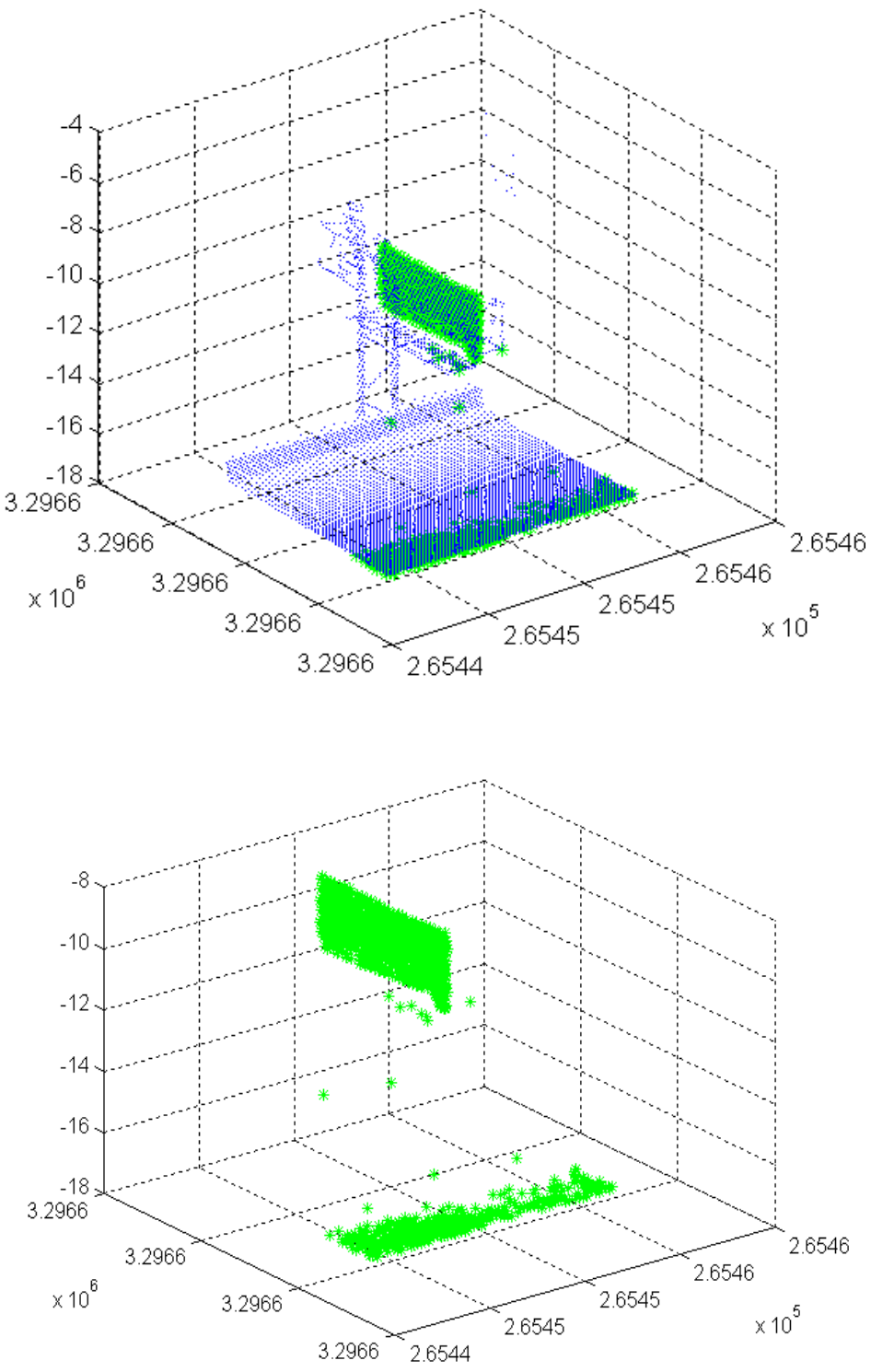


Figure 3-6 Choosing points with I>220 (Green points)

3.4.2 Keeping Points within Planes Facing Driving Direction

For each point in the KD-Tree data set we select all the points within r= 0.5 m radius using nearest neighbor search. Then we calculate the normal vector of the plane which best fits those points using linear least squares. After finding the normal vectors we calculate the angle between that normal with the driving direction which in our specific case was the unit vector along x axis (1,0,0). To give better degrees of freedom, instead of choosing points with normal vectors that have a close to 1.0 dot product with the driving direction (parallel to it), we chose points that have a small (<0.05) dot product with the other two axes ((0,1,0) and (0,0,1)). In this way, we could choose how flexible a tilt is along a particular direction.

As an example, we could chose the normal vectors with dot product <0.1 to (0,1,0) and <0.05 to (0,0,1) which means the plane could tilt a bit more horizontally, which is more probable for the signs according to the driving direction. However, a sign can't tilt a lot vertically as it is not acceptable by standard, and we just want to put a bit freedom for wind tilt or scene error. Figure 3-7 shows the dot product result by color (the darker blue points are more correct as a sign).

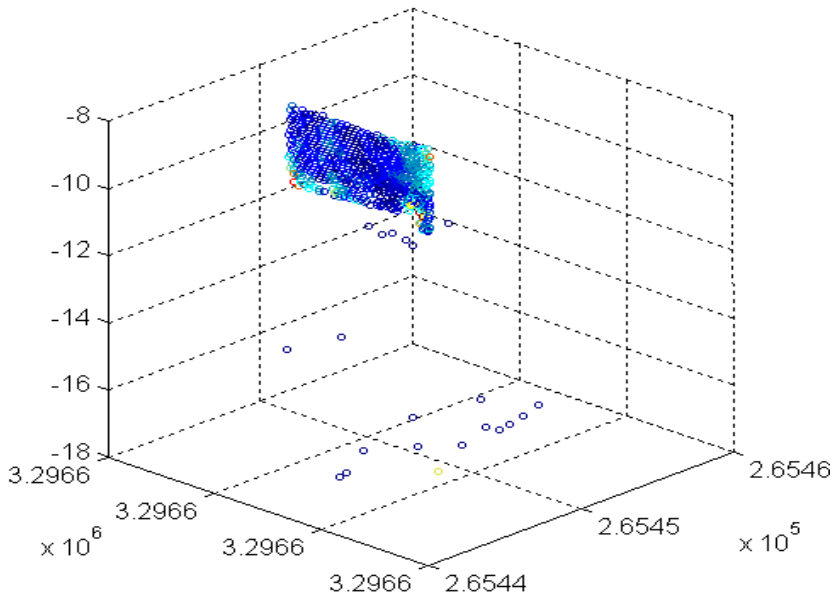


Figure 3-7 Sum of Dot Products to y and z unit vectors
The darker blue points have better normal

3.4.3 Clustering Based on the Neighborhood

After choosing the points with suitable normals, the next step is to get rid of points that are noise or outliers. This can be easily done by a neighborhood search and deleting the points that have less than (for example) 20 neighbors within a certain radius (here we used 0.25 m). Figure 3-8 shows this filtering result.

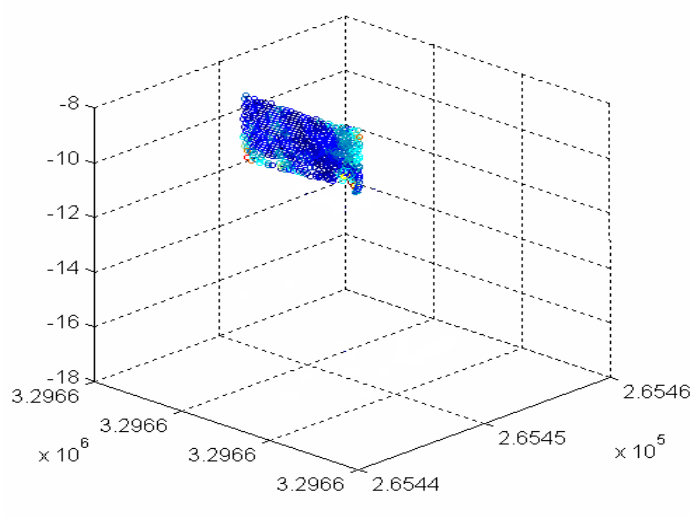


Figure 3-8 Getting rid of outliers

Next step is to find separate clusters. Here, we started with the first point in the filtered data set and applied the region growing algorithm based on only the neighborhood (not the normals). Figure 3-9 shows some clustering results.

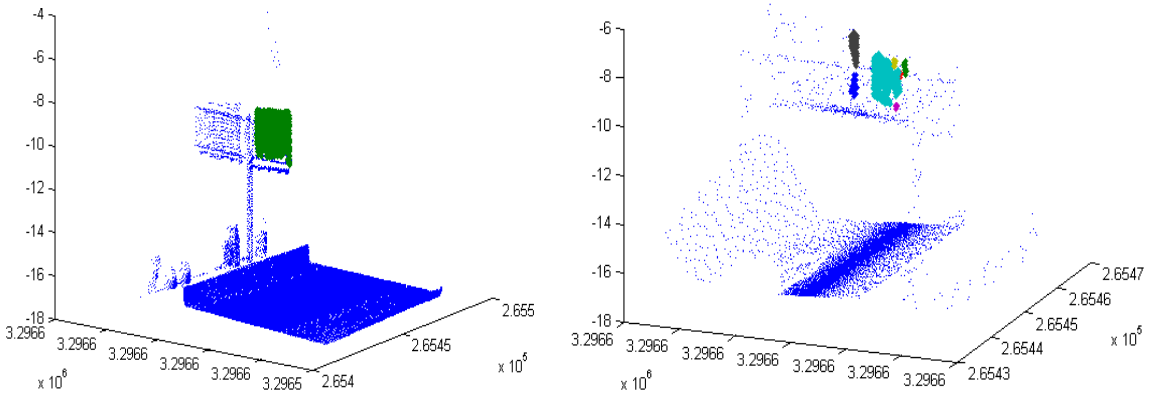


Figure 3-9 Clusters based on the filtered data

3.4.4 Geometry Fitting

After finding the regions (clusters) of interest, we use region growing on the original point cloud to grow the clusters. This allows us to get the correct geometry of the sign even in the case of weathering effects that cause reflectivity lost (for example look at Figure 3-9 *right*).

After obtaining the correct regions we have to fit the suitable shapes to the regions' boundaries. Since the traffic signs have specific geometry we can have a limited set of acceptable shapes (like rectangle, triangle, octagon, or circle). By using the boundary extraction method described in [19], we can fit these suitable shapes to the boundaries, using shape contexts as in [20]. (Figure 3-10)

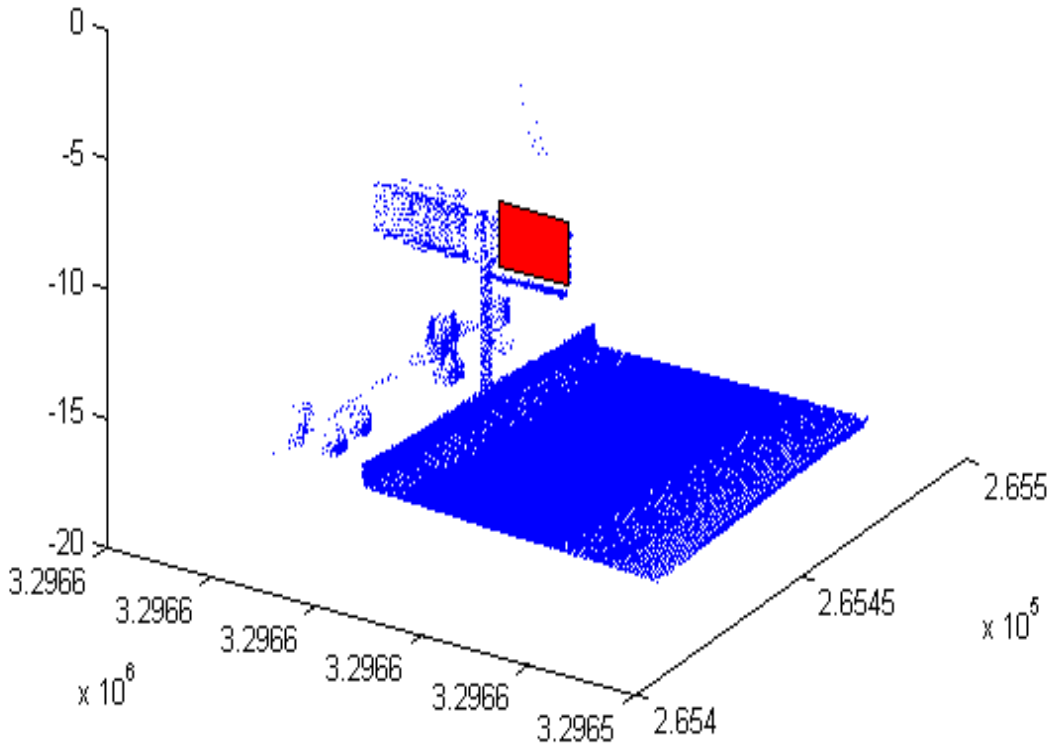


Figure 3-10 Geometry Fitting

3.5 3D Cylinder Extraction

We used Hough transform to estimate the direction of a cylinder. The direction space is constructed by uniformly sampling unit sphere. The method essentially detects the directions along which the points have high density [21]. This method works well for clean cut and non-twisted cylinders (Figure 3-11 *top*). However, when dealing with twisted branches in 3D data, the method doesn't create separable high density areas on the sampling sphere (Figure 3-11 *buttom*). As a result, it worked well for finding the trunk of a tree, because it has the highest hit of points along it. However, the result was not satisfactory for finding branches inside a tree (due to their twists).

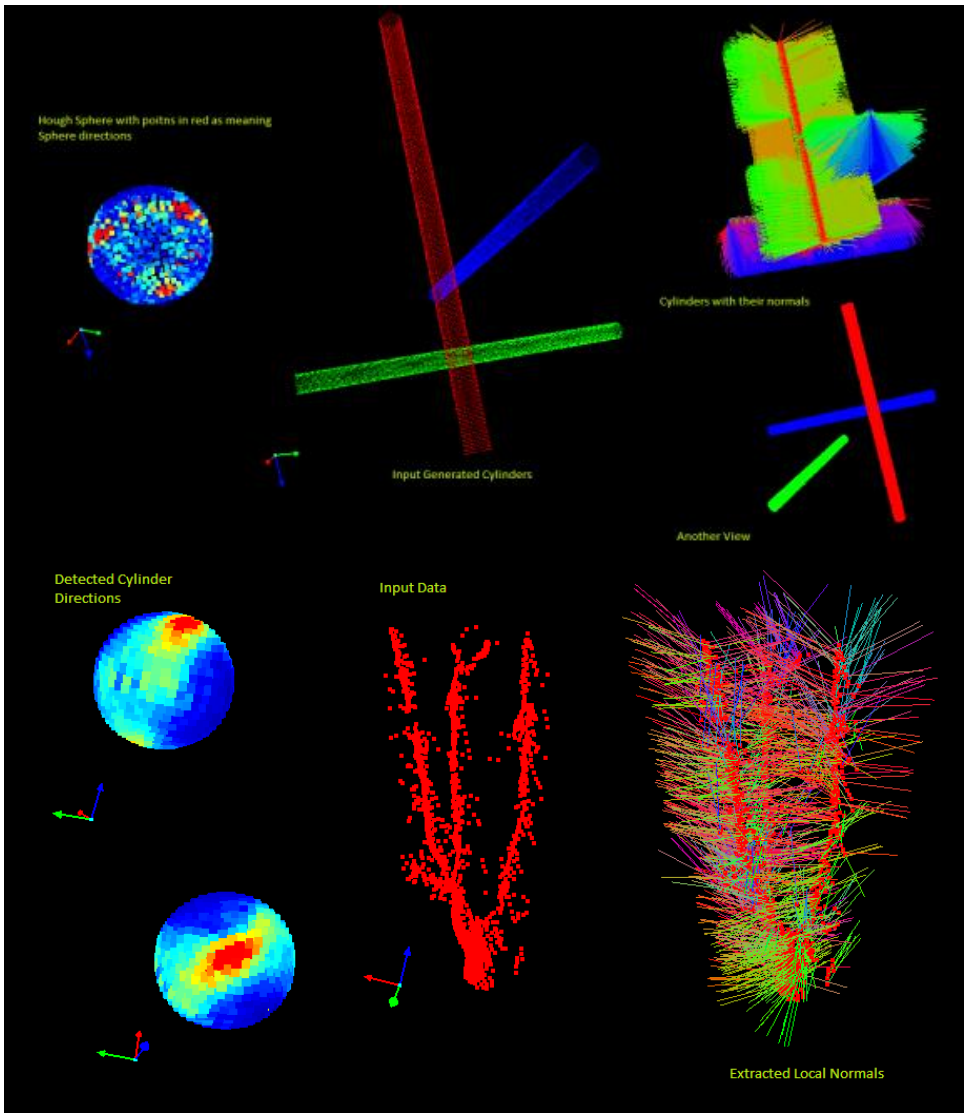


Figure 3-11: Cylinder Extraction using Hough Transform over a Uniformly Sampled Sphere.

3.6 Resolving Parallax Difference Delinquencies in Long Panoramas

As a step towards studying urban scene navigation, we plan to focus on panoramic images of streets. In this section we first explain the problem of *parallax difference*. This phenomenon results in repeated appearance of objects in the far. We propose potential solutions for tackling the problematic issues which mainly concentrates around blurring and hazing of such far objects.

Parallax is an apparent displacement of an object viewed along two different lines of sight (see Figure 3-12). Nearby objects have a larger parallax than more distant objects when observed from different positions. Consequently, if one takes a set of images in a row to cover a street, the objects at the far keep appearing in all these images, although each image contains new objects in the front. Therefore, when these pictures are stitched together for creating a panoramic image, a distant object may repeat multiple times in the panorama. A real example is provided in Figure 3-13, where different colored circles belong to a separate repeated structure.

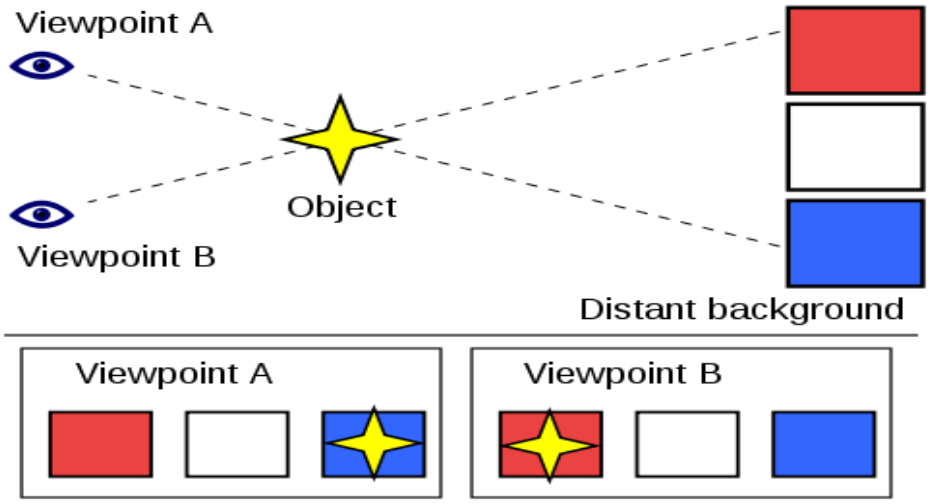


Figure 3-12 When an object is viewed from the point A/B, it appears in front of the blue/red square.

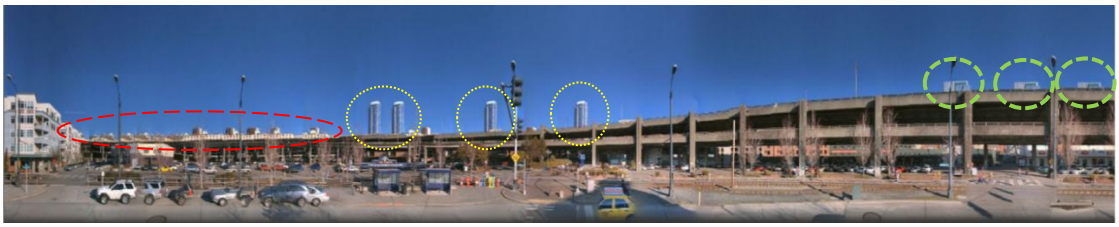


Figure 3-13: Repeated Far Away Structures in Panorama Stitch

Our strategy for coping with parallax difference problem is to reduce the saliency of the distant objects. This is accomplished by blurring or hazing the distance objects, via several steps. Given a clean depth map plus a specific depth of field, one can fuse these two sources via a method called bilateral filtering, to blur objects in the provided depth of field. Unfortunately, clean depth maps are not available in our application and must be estimated from 2D images. Robust depth map estimation from two or more images is another aspect of our problem. The structure of the urban scenes can be exploited to improve depth map estimation. For example, these images are mainly composed of planar surfaces (of building facades). If the facades could be segmented robustly, the result could significantly improve the estimated depth map. In the sequel, we discuss each of these components in greater details.

Our task requires blurring objects in the far of the image, while preserving gradient changes in the scene. One of the most widely used techniques for this purpose is the Bilateral Filtering introduced by Sylvian Paris and Fredo Durand [22]. Figure 3-14 shows a toy example, where the degree of blur varies as the depth of points get farther from a chosen focus range. The depth of focus was chosen by the median of intensity values in the depth image. While the output of this method looks satisfactory to the eye, if the depth of focus is not chosen correctly the resulting output won't be this good. This could, for example, blur the buildings that need to be in focus. Another issue is related to the regions in the image where the estimated depth is not accurate. For example, windows with glasses on them affect the reflective output for the scanner and as a result they mostly appear farther than they actually are. One way to tackle this problem is exploiting the geometric property of facades; they are flat planes. We currently used morphological operations in order to close small holes in facades. The toy examples refined depth (bottom left) shows this correction.

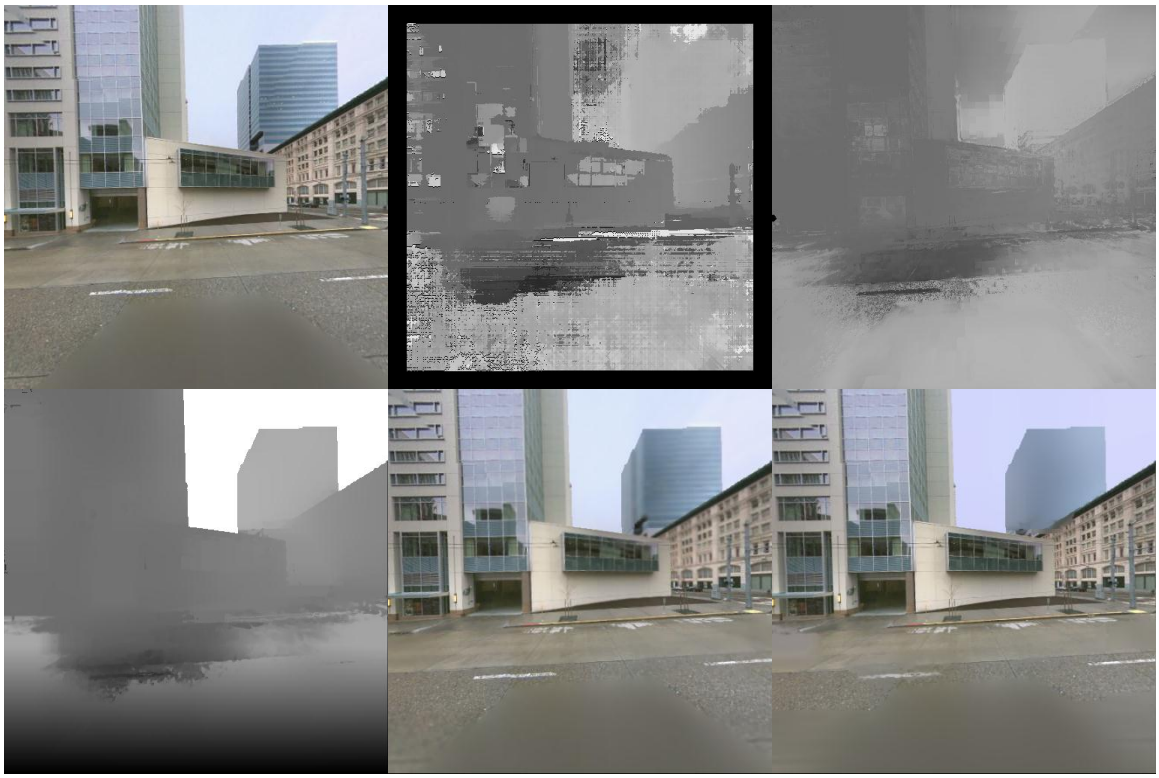


Figure 3-14: Depth Based Blurring of Far Buildings.

Image you want to use your personal camera to take a set of panoramic images (as opposed to relying on laser scanned data). The depth map can be estimated from this set. This is due to the fact that panoramas always consist of at least two images with overlapping regions. This results in a wide baseline stereo problem. Depth estimation under wide baseline assumption is a difficult open problem. Currently, we have tried to warp the images onto each other to reduce the baseline. This allows us to utilize well developed techniques for small baseline matching and depth estimation methods. The depth map obtained by the proposed approach is shown in Figure 3-15.

Figure 3-15: Stereo Matching Based Depth Extraction

As seen in the previous step, the estimated depth map contains a lot of missing regions. In order to fill in these regions, we propose to detect the facades in a given urban image. This way, we know how far we can get from the missing region of a façade such that we do not leave the façade. Then we can use all the available depth values on the façade to interpolate the depth in the missing region. This method comes handy for only urban data where most of the points belong to geometrically well-formed structures.

Currently we are investigating what can be the best approach to segment the facades in the depth images. Similar to the approach explained in section two, we used PCA in a local neighborhood of the points. This allows us to estimate the approximate neighboring plane (normal direction and distance from origin) on each point. We also tried the same technique with local RANSAC on local neighborhoods. The results are shown with RGB color replacement for xyz direction of the plane normal over each pixel in Figure 3-16.

While the wall located on the right looks promisingly purple (meaning it has a consistent plane normal) the rest of the points in the image are not showing consistent output. We are currently trying to extract statistically dominant normal from the scene combined with the robust PCA approach of Yi Ma et all [23].

Once the dominant planes in the scene are identified, we can estimate the focus depth based on the dominant plane in the scene. However, in a long street panorama the focus depth is dynamically changing. As a result we must estimate focus depth in a more enhanced manner. This can simply mean calculating depth at each column of the image independently.

The final step in the problem is to blur the regions in the image which are out of focus. This can be done using the technique mentioned in [24] using the available depth map and focus depth. Since we are choosing the depth of focus column wise, some planes may exhibit abrupt changes in the focus. We need to maintain consistency for blurring by keeping points in the same plane segment in the same focus level. These segments could be extracted by existing façade segmentation techniques [25]. The blurring can be achieved using the aforementioned bilateral filtering technique. We can also choose a color close to the detected sky in the first step to haze the far objects and create a more visually pleasing output.

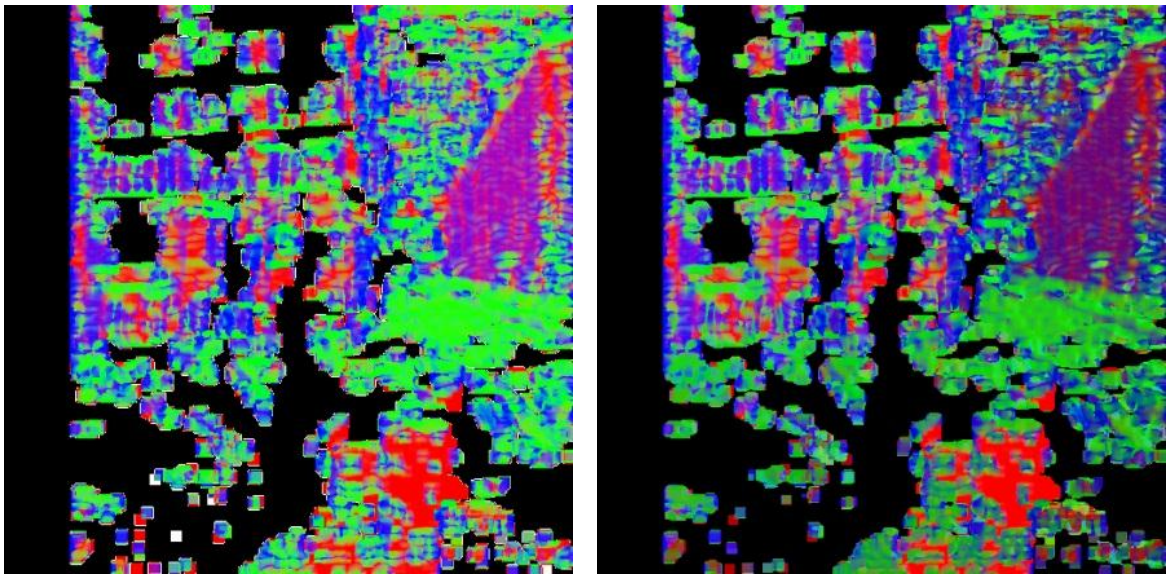


Figure 3-16: Plane Normals in RGB, Left: PCA and Right: RANSAC both on 5x5 neighborhoods.

CHAPTER 4

Meth Morph: Simulating Facial Deformation due to Methamphetamine Usage [26]

We present MethMorph, a system for producing realistic simulations of how drug-free people would look if they used methamphetamine. Significant weight loss and facial lesions are common side effects of meth usage. MethMorph fully automates the process of thinning the face and applying lesions to healthy faces. We combine several recently-developed detection methods such as Viola-Jones based cascades and Lazy Snapping to localize facial features in healthy faces. We use the detected facial features in our method for thinning the face. We then synthesize a new facial texture, which contains lesions and major wrinkles. We apply this texture to the thinned face. We test MethMorph using a database of healthy faces, and we conclude that MethMorph produces realistic meth simulation images.

Methamphetamine abuse contributed an economic burden of $23.4 billion to the United States in 2005 [27]. The meth epidemic has affected large portions of rural, suburban, and urban America [28]. In 2005, fifty percent of the Montana jail population was incarcerated for meth usage or other meth-related offenses [28]. The Meth Project is one of the world's largest anti-meth organizations, and its efforts span eight U.S. states. The Meth Project deployed a large-scale marketing campaign designed to discourage meth use among teenagers in Montana [28]. This marketing campaign uses many images of methamphetamine abusers' faces [29]. As shown in Figure 4-1, these images illustrate common side effects of meth, such as wrinkles, severe weight loss, and lesions. The Meth Project found that the use of such images in anti-meth marketing campaigns can deter teens from using methamphetamines. While teen methamphetamine usage remained relatively constant across the nation from 2005 to 2007, campaigns by The Meth Project helped to reduce meth usage by forty five percent over the same time period in Montana [28].

“Meth will make you look different from normal” serves as one of the key messages of The Meth Project [28]. So far, The Meth Project has illustrated this message with photos of meth users' faces. The leaders of The Meth Project expect that they could further reduce meth usage if it were possible to simulate what a drug-free individual would look like if he or she became addicted to

meth. Toward this goal, we have collaborated with The Meth Project to develop MethMorph, an automated system to produce realistic methamphetamine facial simulations. MethMorph takes images of faces as input and produces MethMorphed images as output.

Developing an automated system to produce methamphetamine facial simulations presents three key challenges. The first of these challenges is to recognize the overall facial region, as well as the chin line, mouth, eyes, and nostrils in an image. Once the facial features are detected, the next challenge is to warp the shape of the face to depict wrinkles and significant weight loss. The final challenge is to depict lesions by changing the texture and color of the warped face. We address the challenges of facial feature detection, facial deformation, and facial texture modification in Sections 2, 3, and 4, respectively. In Section 6, we conclude that MethMorph produces realistic methamphetamine facial simulations.

Figure 4-1: Representative before-and-after images of a meth user. Courtesy of Faces

4.1 Facial Feature Detection

Facial feature detection is crucial for enabling a realistic, automatic methamphetamine facial simulation. Within the facial region, we detect the location of the chin line in order to know what portion of the image to deform into a thinner, more wrinkled face. Chin line detection is particularly important for deformation, because it allows us to thin the cheeks without unintentionally thinning the neck. In addition, detecting the locations of the eyes and nostrils

allows us to avoid placing lesions on the eyes and inside the nostrils. Figure 4-2 illustrates some of our facial feature detection goals.

Viola and Jones introduced a face detection method that uses a cascade of simple [30], [31]. Given an image, the classifiers quickly reject image regions that do not even slightly resemble a face. The Viola-Jones method then explores remaining image regions with additional classifiers. After applying several classifiers, the method concludes where the faces, if any, lie in the image.

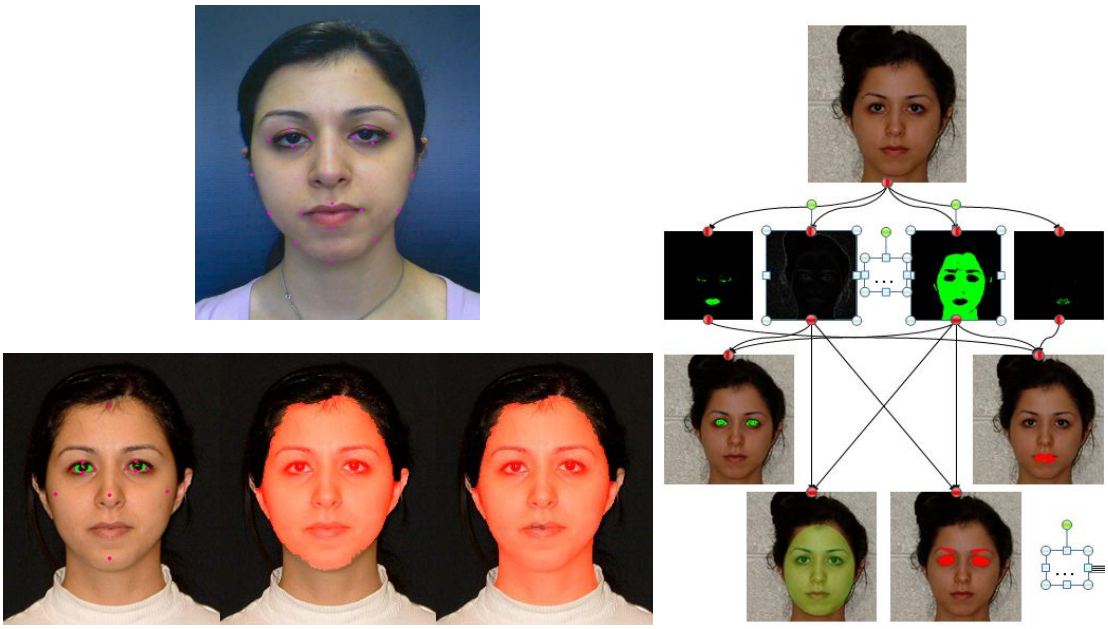


Figure 4-2: Facial Feature Extraction. We also detect the nostrils, though nostril detection is not shown in this figure

4.2 Related Work in Facial Feature Detection

Over the past decade, the computer vision community extended the Viola-Jones cascade to create numerous face and facial feature detection methods. Castrillon et al. survey and evaluate several of these Viola-Jones based methods for localizing faces and facial features in images in [32] and [33]. Many of these methods are integrated into the Open Computer Vision Library (OpenCV) [34]. The work by Castrillon et al. influenced our choices of methods for detecting the face, eyes, nose, and mouth in MethMorph.

4.3 Face Detection

The facial feature detection methods that we present in Sections 2.3 through 2.6 require the face to be detected accurately. Prior to detecting facial features, we first localize the face within an image. Since our facial feature detectors rely on the face being localized properly, it is crucial for our face detection scheme to perform well for a variety of faces.

There exist numerous Viola-Jones based cascades for facial region detection. A recent study by Castrillon et al. [32] found that two cascades called FA1[5] and FA2[6] developed by Lienhart et al. [35] offer greater accuracy than other publicly available cascades in detecting the facial region. The cascades by Lienhart et al. [35] were among the first Viola-Jones based cascades to appear in OpenCV [34].

Given an image, we localize the facial region with the FA1 cascade developed by Lienhart et al. [35]. Our experimental results show that the face detection succeeds for 100% of the twenty-seven images that we tested.

4.4 Robust Chin Line Detection

Once we localize the facial area, we then extract the exact position of the chin line. For chin line detection, we use a method called Lazy Snapping [36].

Lazy Snapping, which combines GraphCuts and over-segmentation, has minimal dependence on ambiguous low-contrast edges. Thus, Lazy Snapping reduces the dependence on shadows for chin line detection.

[5] FA1 is a common abbreviation for haarcascade frontalface alt
[6] FA2 is a common abbreviation for haarcascade frontalface alt2

4.5 Eye and Eyebrow Detection

We localize the eye region with a detection cascade developed by Bediz and Akar [37]. This detector is implemented as part of OpenCV [34]. Out of all publicly-available Viola-Jones cascade eye region detectors, Castrillon et al. found that the Bediz-Akar detector offers the greatest accuracy [32]. Within the eye region, we apply Canny edge detection [38] to locate the corners of the eyes.

Our eye detection method succeeded in 100% of our twenty-seven input images. Figure 4-3 illustrate successful eye detection. Note that Figure 4-3 also show eyebrow detection, but eyebrow detection is not essential to our MethMorph system.

4.6 Lip Detection

We localize the mouth region with the ENCARA2 lip detection cascade [39]. Castrillon et al. found that, out of all publicly-available Viola-Jones cascade mouth region detectors, ENCARA2 offers the greatest accuracy [32].

Within the mouth region, we apply a time-adaptive self-organizing map (SOM). A study by Kamali et al. showed that this time-adaptive SOM exhibits greater accuracy than Snakes [40] for localizing the boundaries of the lips [41]. This method succeeded in localizing the lips in all of our tests.

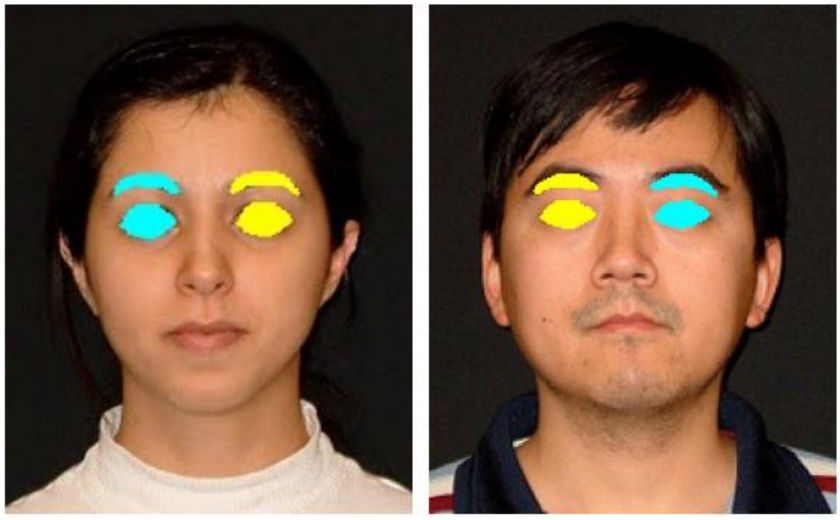


Figure 4-3 Successful eye detection

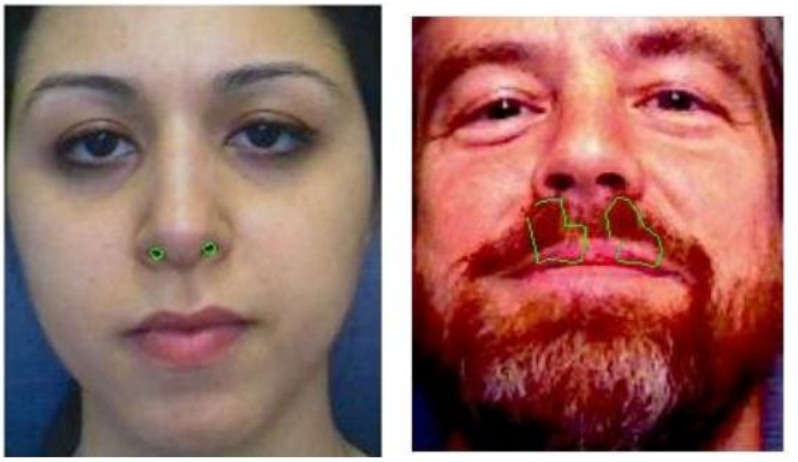
4.7 Nose and Nostril Detection

We localize the nose region with the ENCARA2 nose detection cascade [39]. Out of all publicly-available Viola-Jones cascade nose detectors, a recent study found that ENCARA2 o ers the greatest accuracy [34].

Within the nose region, we apply Snakes [40] to obtain a precise outline of the nostrils. The combination of the ENCARA2 cascade and Snakes successfully localized the nostrils in 93% of the twenty-seven faces on which we tested MethMorph. Figure 4-4 *(a)* shows the typical accuracy of our nostril detection technique. In Figure 4-4 *(b)*, a mustache adds additional clutter, which reduces the likelihood that Snakes converge to the nostrils.

4.8 Facial Deformation

Once we detect the face, eyes, chin line, nostrils, and lips, our next objective is to thin the face. To deform the facial thinning, we apply a method developed by Fang and Hart for deforming the surfaces of objects in images [42]. The Fang-Hart deformation method takes an image along with an existing curve in an image, curve1, and an additional curve, curve2. The method then deforms curve1 to the shape of curve2. In addition, the Fang-Hart deformation method adapts the texture from the original image to keep its original proportions, while reshaping the texture to fit onto the deformed image. To achieve this deformation and retexturing, the Fang-Hart method extends techniques such as GraphCuts, Poisson image editing [43], and Image Completion [44].



(a) Properly-detected nostrils (b) Mustaches make nostril detection more difficult

Figure 4-4 Nostril detection examples

In the context of MethMorph, the chin line detected in Section 2.3 serves as curve1 in the Fang-Hart deformation. We compute curve2 as the average of the chin line, curve1, and a custom-made thin face template. Then, we apply the Fang-Hart deformation method to deform curve1 into the shape of curve2. The result is a thinned face that is characteristic of meth users. We find that, for all the images that we tested, the Fang-Hart method succeeds in producing a realistically thinned face.

4.9 Lesion Simulation

Given a deformed image in which facial feature locations are known, the final step is to replace the healthy skin texture with a texture that has the lesions that are characteristic of meth users. MethMorph's lesion simulation method is based on Textureshop [45], a technology developed and patented [46] by the University of Illinois for retexturing surfaces depicted in a single raw photograph.

What differentiates Textureshop from other retexturing approaches is that Textureshop avoids the reconstruction of a 3D model of the shape depicted in a photograph, which for single raw un-calibrated photographs is a notoriously error prone process. It instead recovers only the orientations of the depicted surfaces and uses these orientations to distort a newly applied texture so it appears to follow the undulations of the depicted surface. While Microsoft and Adobe have already expressed interest in licensing the patented Textureshop technology for their own software, The University of Illinois is allowing MethMorph to serve as its premiere application debut.

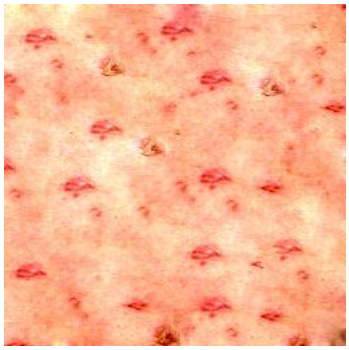


Figure 4-5 Facial lesion texture template

In the general case, Textureshop works as follows. First, use a reflectance model to compute the normals of surfaces in an image. Next, group the surface into patches that have similar surface

normals. Then, extract and store the textures from these surfaces. Finally, apply a new, custom synthesized texture to surfaces (facial skin, for example) in an image. For one or more objects in the resulting image, a new, synthetic texture follows the underlying objects.

For MethMorph, Textureshop uses Poisson image editing [43] to synthesize a new "skin with lesions" texture based on the original skin texture and our custom-made lesion texture template, which is shown in Figure 4-5. The result is a face that has realistic lesions and also retains its original skin tone.

Note that we must avoid placing lesions on the eyes or inside the nostrils. To achieve this, we use the eyes and nostrils that we detected in Sections 2.4 and 2.6. Also, since the facial skin color often does not match the skin color of the lips, we also separately blend the original lip texture with the lesion texture. To differentiate between the facial skin and the lips, we use the lip detection results from Section 2.5.

4.10 Results

Figure 4-6 illustrate the success of MethMorph in producing realistic methamphetamine facial simulations. The success of MethMorph relies on accurate facial feature detection. Given the high success rates of the facial feature detection methods presented in Section 2, it is no surprise that MethMorph offers a high end-to-end success rate.

4.11 Conclusions

MethMorph applies several recent innovations in computer vision and graphics. We found that Viola-Jones based detectors are sufficient for localizing facial feature regions. The application of snakes to the nose region offers a high success rate for precisely detecting the nostrils. The Fang-Hart deformation method succeeds in thinning the face in the manner that a meth user would experience. The Textureshop retexturing technology enables the application of lesions to the skin.

In conclusion, MethMorph succeeds in simulating the visual appearance of meth addiction. Our work also demonstrates the effectiveness of Fang-Hart Deformation method and Textureshop in a real-world application. MethMorph is slated to play a significant role in The Meth Project's campaign against meth usage in the United States. Thus, MethMorph offers significant value to society while also demonstrating the effectiveness of recent computer vision and computer graphics methods.

(a) Healthy Face (b) MethMorphed Face

Figure 4-6 MethMorph applied to a healthy face succeeds in thinning the face in the manner that a meth user would experience.

CHAPTER 5

Robust Classification of Curvilinear and Surface like Structures in 3D Point Cloud Data

Analysis and classification of 3D point cloud data is an important component of applications such as map generation and architectural modeling. Point clouds acquired through laser range-scanning are typically very noisy, making analysis difficult. In this paper, we propose a novel method to cope with such noisy data. The main application of our work is the classification of the structures within a point cloud into curvilinear, surface-like and noise components. The measures taken to ensure the robustness of our method generalize and can be leveraged in noise reduction applications as well. The proposed classification method relies on a combination of edge, node, and relative density information within an Associative Markov Network framework. With this method, we are able to robustly extract complicated structures such as tree branches. We compare our work with another state of the art classification technique called Directional Associative Markov Network (DAMN) and show that our method can achieve significantly higher accuracy for the classification of the 3D point clouds.

In the past decade, robust classification of point cloud data has been a topic of great interest, particularly in navigation environments and forest inventories. The main reason for this interest is due to the wider availability of 3D laser scanners in our times ([47], [48]). Classification of data in such scenarios is a very challenging task for two key reasons. Firstly, trees (and many architectural structures) lack a well-defined geometry when viewed globally, making difficult the learning of a model or prototype from given data. Secondly, diverse geometrical structures stand out at proper local scales (for instance, a branch looks like a cylinder), and therefore, it is easy to make wrong assessments if one does not take in consideration the scale-hierarchical nature of the local arrangements of these structures.

One typical approach for fast data labeling is applying a local feature space clustering method such as Mean-Shift, K-Means, or X-Means on the data. These techniques obviously

suffer from the limitation of using only local statistics which can easily lead to incorrect inference. In our early experiments, such techniques created entirely inadequate classifications.

A more refined approach consists in using both local features and neighbor- hood consistency for adequate point cloud classification. A representative tool, widely used for this purpose, is a graphical model called Associative Markov Network (AMN) [49], [50], and [51]. Yuri Boykov et al. [49] survey how to use Min- Cut/Max-Flow algorithms for solving these graph based classification problems. While the surveyed methods focus on solving the min-cut problem itself, they do not consider what features are most suitable for the particular purpose of point cloud structure classification.

In general, AMN is a graphical model that is deliberated on a point cloud in order to associate each of its nodes (points) with a label. This classification process not only considers features associated with each node, but also reflects how well the guessed label of the node is consistent with its neighbors. As of to- day, there are quite few studies on the construction of a suitable graphical model for point cloud data so that this model can achieve an accurate classification. This is specifically the focus of our paper. We examined several features of the point cloud data and found their best combination for robust classification of connected structures within this data using AMNs.

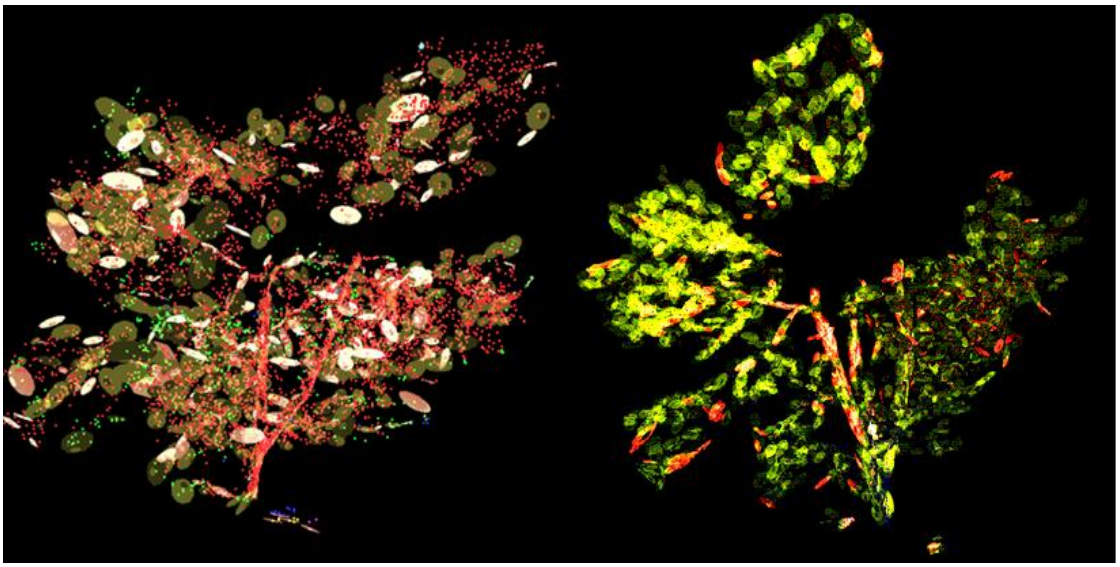
Our initial focus was on the classification of branches versus leaves in data from laser scanned trees. The initial results showed that an AMN constructed from neighborhood density and angular information on edges leads to a good structural classification for tree data. After a deeper investigation, we found a technique good for classification of curvilinear and surface-like structures in general point cloud data.

Local features that are most commonly used for classification of point cloud data are point intensity, point coordinates, neighborhood density, and principal component analysis (PCA) on a local neighborhood [52]. All the AMN based methods mentioned in this paper use one or a combination of these features as local information in order to perform the classification. Needless to mention, if a local feature is computed inaccurately, these algorithms fail to generate a viable classification.

Another challenging problem that AMN based methods face is choosing a correct neighborhood size for local computations. Unnikrishnan et al. [47] and Lalonde et al. [53] describe a method for choosing the best neighborhood size for calculating the normal at each

point on a surface. They indicate that classification based on local features is more accurate when using PCA information extracted at the proper scale. However, this method does not work well on chaotic structures like trees. Figure 5-1 *(left)* displays this problem by using red versus green labeling for points belonging to locally cylindrical structures (branches) versus noise (leaves). Using their model, most of the points are labeled as cylindrical. The fixed scale method results in a much more accurate classification. (Figure 5-1 *right*)

Munoz et al. [50] use Directional Associative Markov Networks (DAMN) in order to extract structures that have a well-defined direction. This approach can- not be used for the extraction of the branches of a tree since these structures do not follow a specific global direction. Moreover, even if there are only connected structures that follow a finite set of directions, the DAMN approach cannot extract connected structures with multiple different directions (for instance, a twisted tree branch).



(a) Variable scale PCA (b) Fixed scale PCA

Figure 5-1 Local PCA for the branch (red-oval) vs. leaf (green-sphere) classification

Xiong et al. [54] and Munoz et al. [55] create newer subcategories of AMN which hierarchically take advantage of the contextual information to further enhance the classification. However, the mentioned technique fails when dealing with a twisted linear or curved surface which does not follow a specified global orientation.

Shapovalov et al. [56] use a method that partially relies on AMNs for extracting connected segments in point clouds. While their work shows promising results, it is not able to classify data on arbitrary bases.

As discussed in this section, the existing AMN based techniques do not fo- cus on choosing good features for extracting connected structures from point clouds, such as tree branches or other curved natural structures. In this paper, we investigate several different combinations of local features to leverage the AMN technique for robust classification. Furthermore, we enhance the numerical stability of this technique. We present our method in detail in the consecutive sections.

5.1 Data Classification Using Markov Networks

In this section, we formally state the classification problem and show how an AMN based classifier can be used to generate a solution.

5.1.1 Problem Description

As described in [50], we have a set of n random variables $Y = \{Y_1, \ldots, Y_n\}$, and a scoring function. The goal is to find the assignment of values of $y = \{y_1, \ldots, y_n\}$ to Y that maximizes our scoring function. In our problem, Y_i 's are the 3D points and, the values they can take correspond to their labels. Each point has a set of corresponding features, x, which are extracted from the data (like geometry, color, intensity, PCA). The effect of the observed features on the labels is modeled by the conditional distribution $P_w(y|x)$, which is parameterized by a vector w. Note that y is a vector of length n, thus it is joint distribution of all node labels that is conditioned on all the features. In the following subsections, we explain how the distribution $P_w(y|x)$ is modeled for our task and how w is estimated. Estimation of w can be formulated in terms of a supervised learning problem. Therefore the classification will have two main steps: 1) learning w from already labeled data, and 2) inference for a new non-labeled scene based on the learned model $P_w(y|x)$.

5.1.2 Associative Markov Networks

An Associative Markov Network (AMN) defines the joint distribution of random variables Y_i conditioned on the observed features X. The problem is associated with a weighted graph, where the random variables Y_i (each variable is a 3D point in our problem) correspond to nodes and each edge defines the interaction between each variable with its surrounding nodes. Each node has potential $\phi_i(y_i)$ and each edge has potential $\phi_{ij}(y_i,y_j)$. The potentials convert the features to numerical scores indicating their affinity to the labels. Construction of this graph (including the way potentials are defined) is task dependent and non-trivial. For our task of interest (classifying tree leaves from branches in point clouds), we propose to build this graph by running an ε-ball neighborhood search on each node (3D point) and create an edge between that point and whatever that falls inside the ball.

The dependence of the potentials to their features $x=\{x_i,x_{ij}\}$ is modeled to be log-linear, where $x_i \in \mathbb{R}^{d_n} = node\ feature$ (such as 3D position, intensity, color, eigen vectors) and $x_{ij} \in \mathbb{R}^{d_e} = edge\ feature$ (for example, the angle between the dominant eigenvectors on the nodes). Therefore, when to a node is assigned a label k, the log of the node potential is modeled as:

$$\log \phi_i(k) = w_n^k . x_i$$

In the Markov Network, a variant of the Pott's model is use, where the assignment of differently labeled nodes that share an edge is penalized and assignment of nodes with same labels is accepted. Therefore:

$$\forall k \neq l \quad \begin{cases} \log \phi_{ij}(k,k) \geq 0 \\ \log \phi_{ij}(k,l) = w_e^{k,l} . x_{ij} = 0 \end{cases}$$

We also need to mention that the weights and features are constrained to be non-negative. The log of the joint conditional probability function is:

$$\log P_w(y \mid x) = \sum_{i=1}^{N}\sum_{k=1}^{K}(w_n^k, x_i) y_i^k + \sum_{(ij)\in E}\sum_{k=1}^{K}(w_e^{k,k} . x_{ij}) y_i^k y_j^k - \log Z_w(x)$$

$$where \quad Z_w(x) = \underbrace{\sum_{y'} \prod_{i=1}^{N} \phi(y'_i) \prod_{ij\in E} \phi_{ij}(y'_i, y'_j)}_{Partition\ Function}$$

Finding the assignment which maximizes this probability is, in general, an NP-hard problem [50]. However, with tasks involving only two labels, it is possible to compute the solution

efficiently [49]. While this paper works with a 2-label problem (branches vs. leaves), we mention that extending the problem to cover more labels is still possible if approximate solution are tolerated. One example to achieve such approximation is α-expansion [53]. For details on estimating the parameter vector *w*, please refer to [54]. We follow the sub-gradient method for the maximizing the score function, as explained in [50].

5.2 Geometry Driven Associative Markov Networks

Munoz et al. [50] assert that regular AMNs do not produce promising classification because of not considering global coordinate directions. The authors utilize position and principal direction at any point of the cloud as features of that point. They claim that, although direction is a crucial piece of information for classification of the point data, it should not be directly used in the AMN. They suggest that, when there is no dominant direction in the neighborhood of a point, AMN only relies on a spatial coherence assumption and labels all the nodes in that neighborhood as the same class.

The assumption of clean outstanding direction in a neighborhood is not met in most LiDAR datasets due to noise. In order to cope with this problem, [50] modifies the original AMN to reduce its sensitivity to local directions and lever- ages global directional information instead. This is achieved by first quantizing the set of all possible directions and then considering the points from the entire cloud that have the same discretized direction to be in the same class. This is useful when the structure of the objects consists of a few dominant directions. Examples given in [50] include tree trunks and poles that are mainly vertical structures or power lines that are mainly horizontal structures. However, the assumption of having a few dominant directions does not apply to objects like trees (except for their trunk). The branches and leaves in a tree are quite complex in shape and grow in arbitrary directions. Therefore, in this work, we leverage the continuity of directions to aid the classification process.

5.2.1 Relative Directional Associative Markov Networks (RDAMN)

In this paper, we propose using edge based directional information in a different scheme compared to DAMN. As visualized in Figure 5-2, we use angular difference between points at

the two sides of an edge as edge features for AMN. In Figure 5-2 and Figure 5-3, we show how RDAMN outperforms in separating noise from structure.

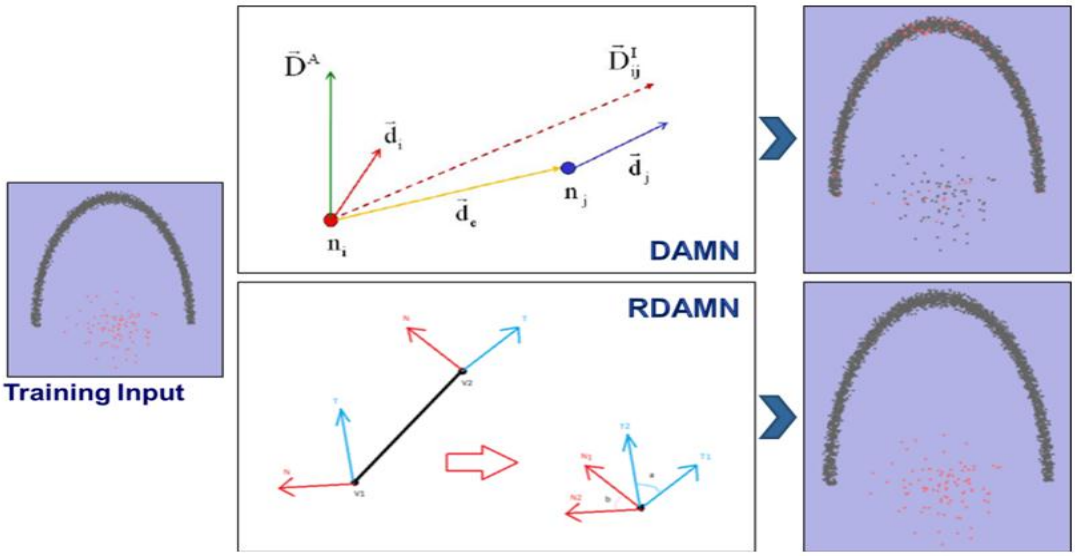


Figure 5-2 Edge Features Comparison, (Left) Structured points in grey and noise in red (Top Middle). D^A refers to global directional bins its difference with edge orientation D^I's is used (more details in [50]), (Bottom Middle) N (red) refers to v_{i0} and r (blue) refers to v_{i2}, (Right) top Inference by DAMN, bottom output of RDAMN

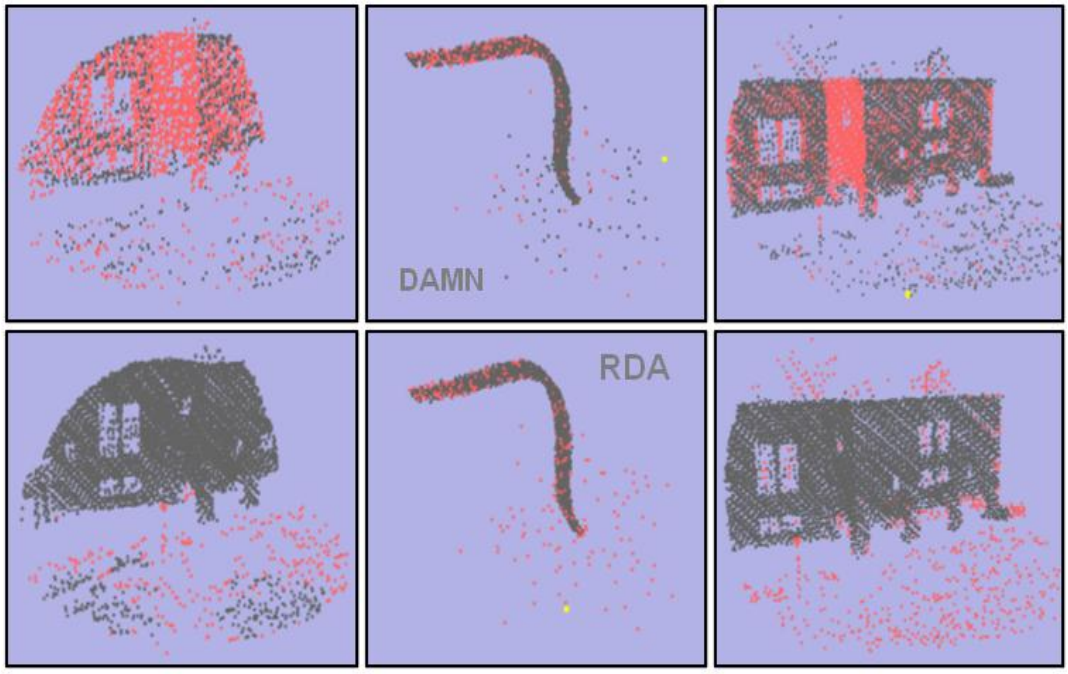


Figure 5-3 Comparing DAMN with RDAMN outcome for extracting skeletal structure vs. noise (Gray color indicates skeleton and Red indicates noise)

5.2.2 Adequate Feature Combination

There are many possible local properties that one can consider within AMN based classification scheme. We have explored the ones listed below:

- *Position:* Such as coordinate values $p_i = (x_i, y_i, z_i)$ for $i = 1,2,\cdots,n$.
- *Intensity:* The amount of light reflected.
- *Local Density* ρ_i: The relative density of an epsilon ball around the i'th point. Below "I" is an indicator function that has value one if the condition is met and zero otherwise.

$$N_i \stackrel{\text{def}}{=} \sum_{k=1}^{n} I_{||p_i - p_k|| \le \varepsilon} \qquad \rho_i \stackrel{\text{def}}{=} \frac{N_i}{\frac{1}{n}\sum_{k=1}^{n} N_k}$$

- *Orientation:* Eigenvectors $v \in \mathbb{R}^3$ and eigenvalues $\lambda \in \mathbb{R}$ for the locally computed covariance matrix C_i. Here

$$n_i \stackrel{\text{def}}{=} \sum_{p_k; ||p_i - p_k|| \le \varepsilon} 1 \qquad \bar{p_i} \stackrel{\text{def}}{=} \frac{1}{n_i} \sum_{p_k; ||p_i - p_k|| \le \varepsilon} p_k$$

$$C_i \stackrel{\text{def}}{=} \frac{1}{n} \sum_{p_k; ||p_i - p_k|| \le \varepsilon} \left|\left|p_i - \bar{p}\right|\right| \left|\left|p_i - \bar{p}\right|\right|^T \qquad C_i v_i = \lambda_i v_i$$

Since $v_i \in \mathbb{R}^3$, we have three eigenvalues $\lambda_{i\,0} \le \lambda_{i\,1} \le \lambda_{i\,2}$. We use $E_i = (\lambda_{i\,0},\ \lambda_{i\,1} - \lambda_{i\,0}\ , \lambda_{i\,2} - \lambda_{i\,1})\ /\ \lambda_{i\,0}$ as (sphereness, surfaceness, cylinderness).

Although there exist several outstanding properties that can be used as features, we need to choose the most relevant subset to avoid over fitting. Following, is the list of the seven combinations that we used on our inputs:

1. E_i
2. E_i and $v_{i\,0}, v_{i\,1}, v_{i\,2}$
3. E_i and ρ_i
4. E_i and ρ_i and $v_{i\,2}^T\ v_{j\,2}$
5. ρ_i and $v_{i\,2}^T\ v_{j\,2}$
6. E_i and $v_{i\,2}^T\ v_{j\,2}$
7. E_i and ρ_i and $v_{i\,0}, v_{i\,1}, v_{i\,2}$

Each of the mentioned features was tested on 18 different point clouds from different scenes. The results indicated that the seventh feature combination was the most successful one; providing a good balance between over fitting and under fitting. Figure 5-4 shows the labeled tree branches / leaves in point cloud data using this choice.

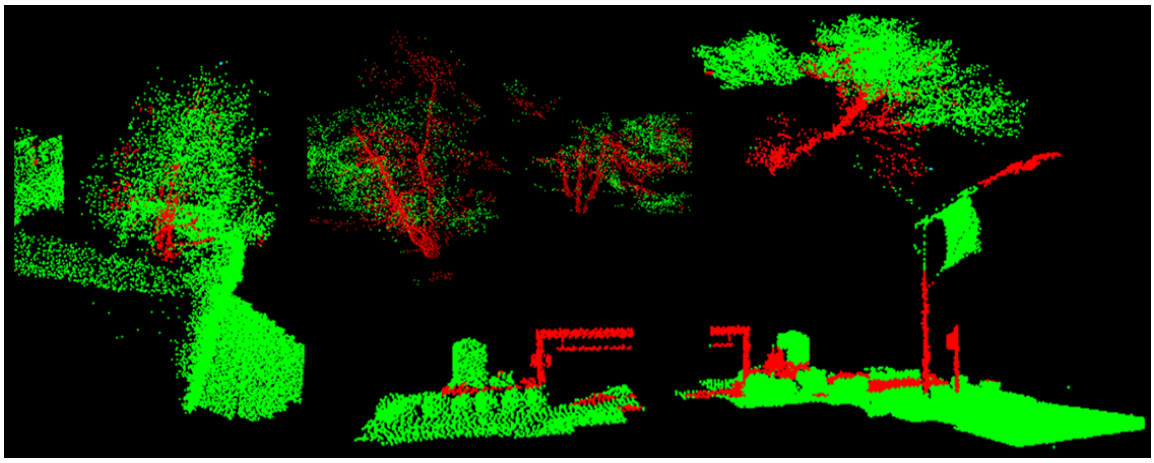


Figure 5-4 Classification results, The points with neighborhoods looking like surfaces or spheres are classified as leaves (green) while points with neighborhoods like lines or cylinders are classified as branches (red).

5.2.3 Logarithmic Barrier

One condition in Markov Random Fields probability model is that the weights vector w must be positive. In [50], this is achieved by replacing negative weights with zero throughout the optimization iterations. However, such sharp change in w affects the numerical stability of the optimization. In fact, we observed convergence difficulty with that implementation. We propose to use the logarithmic barrier method [57] that penalizes weights close to zero. This is a smooth way of maintaining positive weights.

5.3 Experimental Results

In order to test our algorithm on a wide variety of scenarios, we use input data gathered by Velodyne and Sick mobile LiDAR systems and also by Leica static system. Each system has distinct properties. The Velodyne LiDAR data is dense but contains sporadic noise, while the

Sick data is sparser but contains more robust data. We use a KD-Tree data structure to extract epsilon ball neighborhoods of points, computed in parallel and reasonably fast. For principal component analysis, we use GSL's optimized mathematics library for efficient singular value decomposition. Regarding the performance, for labeling roughly between 50,000 to 100,000 points on an Intel core i7 Q720 Windows 7 system, it takes approximately 3 to 5 minutes for learning and 15 to 30 seconds for inference.

For our quantitative experimental comparison, we used sub-sections of data from the 2011 3DIMPVT challenge dataset which was acquired by Leica ScanStation2. It contains a number of accurate range scans in New York City. The dataset contains a mixture of noise points, and human-made structures. We chose a goal of removing noise from structures. We define structures as all the structures that have reasonably a smooth surface. This automatically excludes all the visually scattered vegetation points. We used one dataset as reference data for training both our RDAMN algorithm and the DAMN algorithm [50], and three other datasets as test data. To ensure fairness, we ran a 3 way cross validation by alternating the dataset for which the training occurred. The 3 data sets are hand labeled and hence the learning can still vary a bit from set to set. Table 1 lists the quantitative comparison between DAMN and our method, RDAMN. Figure 5-5 and Figure 5-6 show two samples from the mentioned test data and their statistical results are shown in Table 5-1.

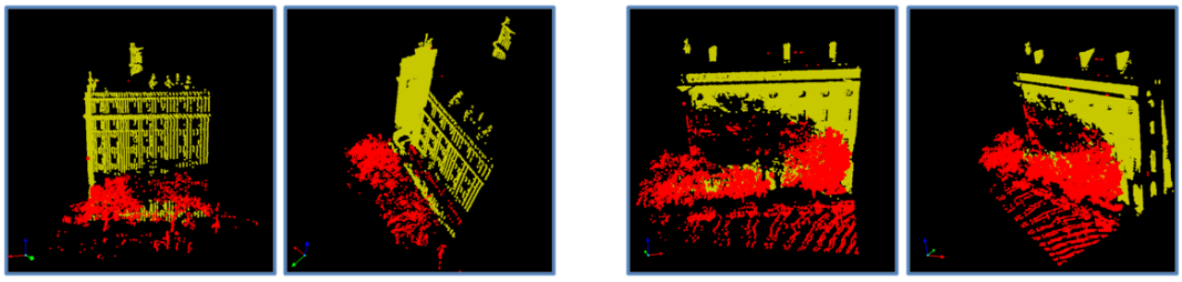


Figure 5-5 ***(left)*** **Data 2** ***(right)*** **Data 4 training input, viewed from two different angles**

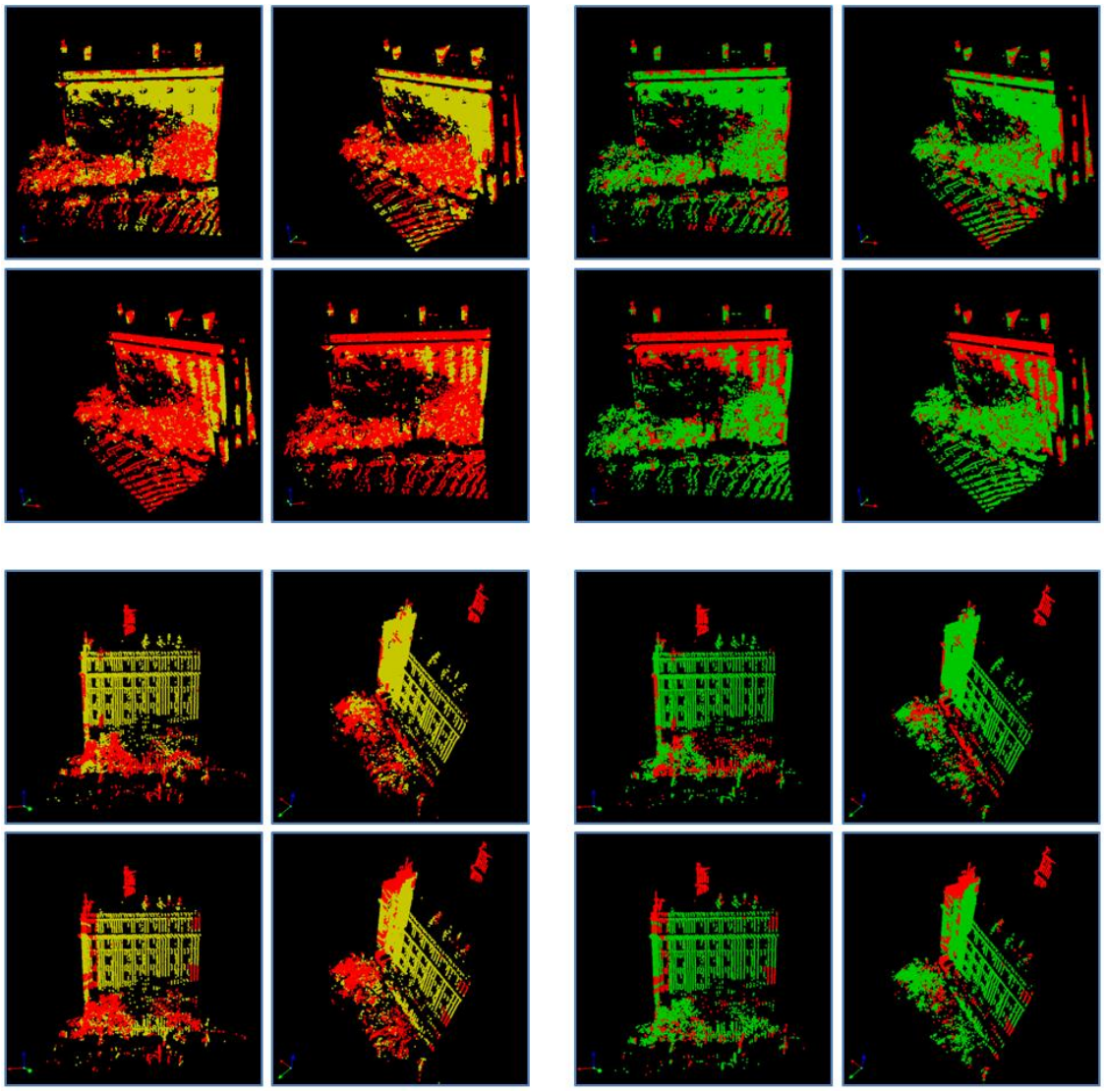


Figure 5-6 ***(left)*** **Algorithm labeling based on learned inputs on Figure 5-5.**
(right) **Green shows correct labeling and red shows incorrect labeling.**
(top two rows) **Data 2 RDAMN on top row and DAMN at the bottom row**
(bottom two rows) **Data 4 RDAMN on top row and DAMN at the bottom row**

Table 5-1 Three way cross validation correct label count RDAMN vs. DAMN

Training Data	Test on	Points Count	Correct Node Count	
			RDAMN	DAMN
Data1	Data2	12291	9819	3852
	Data3	39503	36277	24318
Data2	Data1	32945	21078	13560
	Data3	39503	24430	15339
Data3	Data1	32945	28217	25273
	Data2	12291	10157	5733

5.4 Conclusion

In this paper we introduced a robust technique for classification of LiDAR point cloud data mostly for denoising purposes. Due to consideration of curvilinear flow and neighborhood information in a typical Markov random field, the max flow algorithm is able to robustly learn desired classification and leverage this on the point cloud data. This technique has many real applications particularly in noise removal from LiDAR data for modeling or other industrial purposes.

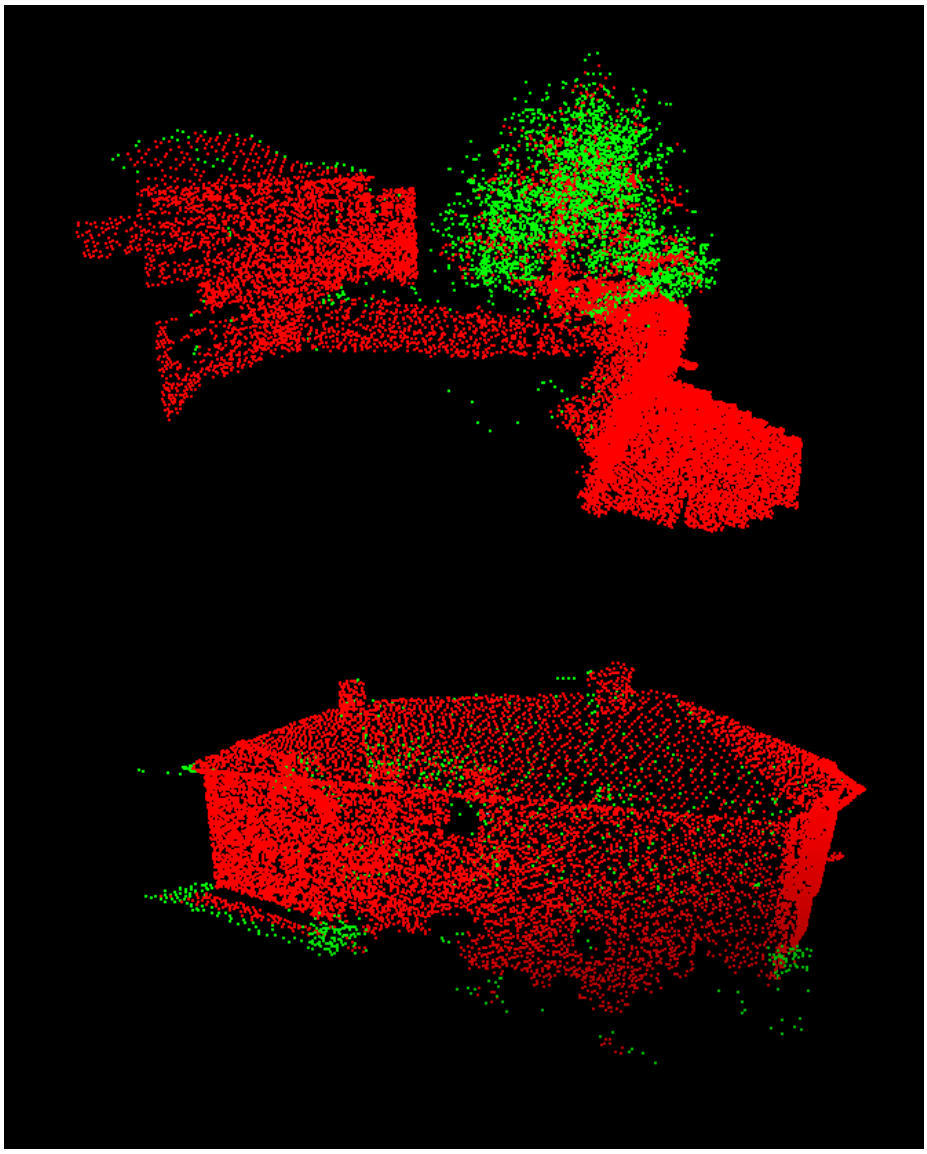


Figure 5-7 Extracting vegetation from laser scanned data using Markov Random Fields.

CHAPTER 6
Linear Clutter Removal from Panoramic Street Images

Panoramic images capture cityscapes of dense urban structures by mapping multiple images from different viewpoints into a single composite image. One challenge to their construction is that objects that lie at different depth are often not stitched correctly in the panorama. The problem is especially troublesome for objects occupying large horizontal spans, such as telephone wires, crossing multiple photos in the stitching process. Thin lines, such as power lines, are common in urban scenes but are usually not selected for registration due to their small image footprint. Hence stitched panoramas of urban environments often include dented or broken wires. This paper presents an automatic scheme for detecting and removing such thin linear structures from panoramic images. Our results show significant visual clutter reduction from municipal imagery while keeping the original structure of the scene and visual perception of the imagery intact.

Multi-perspective panoramic imaging produces visual summaries of scenes that are difficult to capture in a camera's limited field of view. As a result, multi-perspective panoramas have seen increasing popularity in navigation and sightseeing consumer applications. For example, Microsoft Street Slide renders multi-perspective panoramas in real time, thus enabling an interactive urban sightseeing experience [58]. We show an example Street Slide urban panorama in Figure 6-1.

Until automatic multi-perspective panorama production methods were developed, panorama production typically relied on single-perspective, orthographic projections. In single-perspective panoramas, each point in the world is mapped to the closest point in the panorama's plane [59]. As a result, single-perspective panoramas suffer from the unnatural effect that far-away objects and close-up objects appear at the same scale [60]. This effect is particular apparent in long panoramas of city streets. Multi-perspective panoramas avoid this unnatural effect by stitching images from disparate viewpoints in a panorama [61]. Each portion of a multi-perspective panorama looks like a natural-perspective view of the scene, though the panorama as a whole does not adhere to a single linear perspective [62], [63].

Figure 6-1: Panorama of a Long Street [58]

In the last few years, the computer vision community has made significant strides in automating the production of multi-perspective panoramas. In 2004, Roman et al. developed a system that relied on some human interaction to produce multi-perspective panoramas [64]. By 2006, Roman and Lensch succeeded in automating this process [65]. Automatic multi-perspective panorama production involves stitching images together along seams that best merge overlapping features [66]. Toward this goal, stitching techniques prioritize large objects with low depth variances (such as building facades), isolated objects, and objects with small horizontal spans (such as poles and people). However, smaller objects that lie at a different depth can confound stitching, and appear broken or multiple times in the panorama. In Figure 6-2 (Top), the panoramic image shows a smooth stitching of the facades, but power lines which are at different depths are distorted. Figure 6-2 (Bottom) demonstrates how removing *linear clutter* such as power lines enhances the quality of panoramas.

We present a novel method for the automatic removal of linear clutter from multi-perspective panoramas. Our method focuses on the removal of linear features that are situated in front of high-contrast backgrounds, such as power lines in front of the sky. Our method uses a modified Hough transform to detect problematic thin horizontal features. We remove unwanted horizontal features with a short linear filter. These steps form a method that improves the appearance of automatically constructed panoramas. Our method also reduces the amount of user intervention needed for the construction of high-quality multi-perspective imagery.

Figure 6-2 ***(Top)*** **Panorama stitched from a group of images taken along a street, including horizontal line multi-perspective stitching artifacts caused by power lines.** ***(Bottom)*** **The same scene where power line artifacts are removed**

6.1 Background

Methods for automatically detecting and removing wires from images have been developed for outdoor power line inspection and for the cinema special effects industry. In this section, we place our work in the context of past wire detection and removal methods. We also discuss limitations of past work, and we explain how our method overcomes these limitations.

In collaboration with power line maintenance companies, two computer vision studies present methods for detecting power lines in aerial images. These studies enable the use of using small airplanes for inspecting outdoor power lines. Yan et al. apply a Radon transform to extract line segments from power lines in aerial images [67]. Next, Yan et al. use a grouping method to link the line segments and a Kalman filter to connect the detected segments into an entire line. Mu et al. extract power lines from aerial images with a Gabor filter and a Hough transform [68].

The studies by Yan et al. and Mu et al. make the simplifying assumption that power lines are perfectly straight [67].These studies also assume that power lines are made out of a special metal, which has a uniform width and brightness. In contrast, our method breaks image regions into small linear parts that allow power lines to curve, and rely on contrast but not constant color along the line. Therefore, our method succeeds in detecting linear clutter artifacts with varying width and brightness. Also, unlike these power line detection methods, our method both detects and *removes* the linear clutter from images.

Hirani and Totsuka developed a method for removing linear clutter from video frames [69], [70].Their method is especially targeted toward the cinema special effects community. The Hirani-Totsuka method succeeds in applications such as removing wires that actors hang from while doing

stunts and removing scratches in old film. Hirani and Totsuka achieve linear clutter removal by applying *projection onto convex sets* (POTS).The method is effective, but it is not fully automated and requires the user to manually detect the linear clutter in the image. In contrast, our linear clutter removal method is fully automated.

6.2 Linear Clutter Detection

Existing methods for extracting lines from images, such as the methods discussed in Section 2, rely on either the Hough or Radon transform [71], [72]. These line detection techniques alone are insufficient for removing telephone and power wires. First, these wires are usually not straight lines and form catenoid curves. Second, current line detection techniques utilize edge detection output, which for a thin line appears as a pair of edge-detector gradient lines on each side of the wire instead of the wire itself. (We illustrate this in our Experimental Results section and in Figure 6-7.) We customize these edge detection approaches to handle thin, horizontal features. We also consider that the color of the top and bottom neighboring pixels on a linear wire are similar. This criterion further enhances our line detection by making sure the color of the regions on each sides of the line are the same which is contrary to generic edge detection filters. Moreover, we consider that wires can have different diameters so we have to capture them at any width.

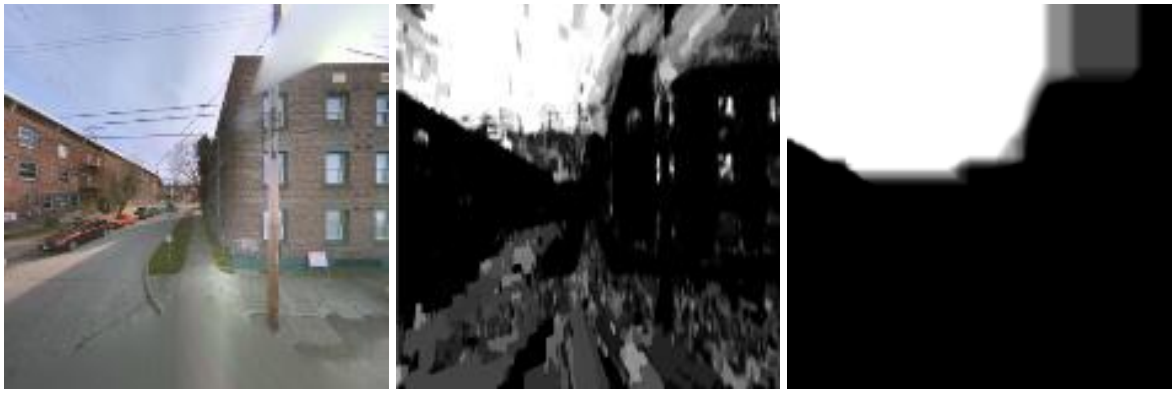


Figure 6-3 Finding the sky region of an image. *(Left)* Original Image. *(Center)* Sky Confidence Map *(Right)* Our Refined Sky Confidence Map.

Due to the visual complexity of building structures, we are less interested in removing lines from front of buildings facades, and focus primarily on thin horizontal occlusions of the sky region. Building façade structures have complex textures that themselves often contain many horizontal lines (like window separators and bricks). We seek to avoid blurring the fine details of these building textures. Therefore, we focus on the more distracting sky related candidate regions for line

removal. Our wire removal algorithm first identifies the region of the image corresponding to the sky, and then tracks and removes thin linear features within this region.

We first characterize the sky. Using the input images for the panorama, we find sky related pixels using a depth map if available [73], Skyfinder [74], or scene interpretation [75]. We then create a 2-D (HxS) histogram of the hue and saturation values of the pixels detected as "sky," and selecting the most popular hue/saturation combinations as the sky color. We illustrate an example sky mask in Figure 6-3.

We then construct a sky mask, where each pixel in the mask is its value from the (normalized) sky histogram for that pixel's hue and saturation. The resulting mask will be noisy and contains many small non-sky regions so we filter it using a Gaussian (or edge-preserving bilateral) low-pass smoothing filter, followed by a morphological "opening" operation consisting of erosions followed by dilations to remove features such as windows reflecting the sky.

For extracting the wire confidence map, we convolve the image with a set of different vertical width filters in order to find the pixels that most likely belong to horizontal lines. We define a family of filters $Filter_1 = [1 \ldots 0 \ldots -1]^T$ and a second family of filters $Filter_2 = [1 \ldots -2 \ldots 1]^T$.

$Filter_1$ searches for pixels whose top and bottom neighbors are similarly colored. $Filter_2$ searches for pixels that are significantly darker than their vertical neighbors. For 512x512 pixel input images, we observed that the number of pixels in both filters best to ranges from 3 through 11 (this range is the parameter which users need to provide before running our algorithm).

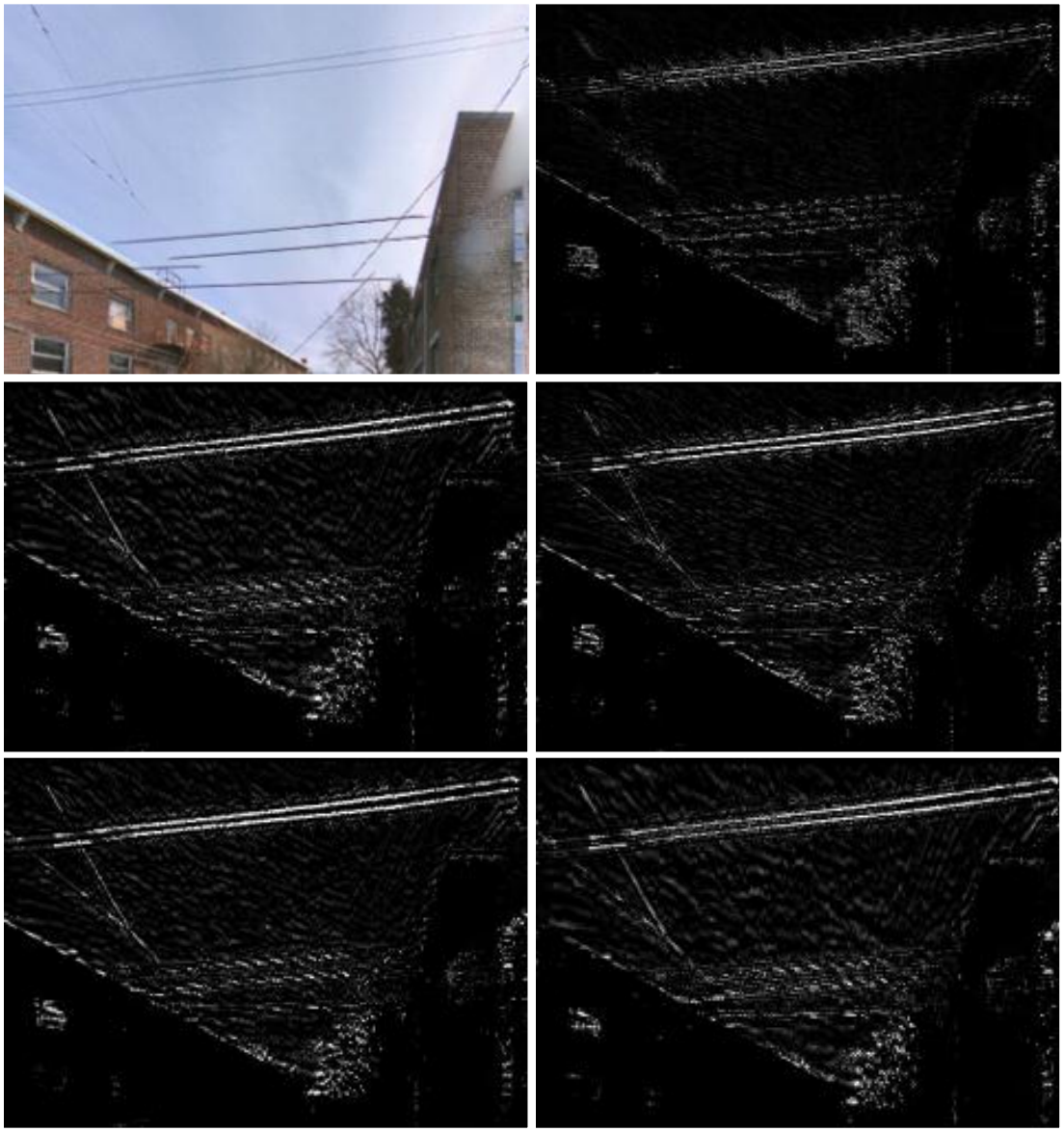


Figure 6-4 *(Top left)* Original image, *(Top center, top right, bottom row)* Line confidence map for filter widths of 3, 5, 7, 9 and 11 pixels

We compute the quotient $Filter^{l}(p_i) = |Filter_2^{l}(p_i) / Filter_1^{l}(p_i)|$ for each filter width 3,5,…,11, and for each pixel p_i in the sky region. We show an example application of these filters in Figure 6-4. For each pixel, we pick the largest absolute value returned from all filter sizes and scale the result by the sky region confidence map, $Filter(p_i) = \max_{l\in(3,5,\ldots,11)}(Filter^{l}(p_i) \times HS(p_i))$. Two variables called min_line_width and max_line_width (in our example 3 and 11) need to be provided by the user.

Using a generic Hough transform, some pixels will be detected that don't belong to horizontal lines. We modify the Hough transform to find candidate partial horizontal lines in the image. We remove these pixels by considering the gradient entropy at each pixel. Pixels which belong to a gently curving line should have low gradient direction entropy, so we remove pixels with high gradient direction entropy. This can easily be done by passing a smoothing filter over an image of the gradient directions$\theta = atan(\frac{grad_{vertical}}{grad_{horizontal}})$ of the input image of potential lines.

We create four bins for angles (0-45), (45-90), (90-135) and (135-180) degrees that are incremented when a pixels gradient falls within that range of directions. If the entropy of a bin is above 80% of the maximum entropy value for a line (since we have 4 bins the maximum line entropy is about 1.39 [76]) this means this region belongs to a non-consistent gradient (clutter) so we remove it.

Line segments near boundaries of sky regions can be missed by this classifier. Hence, our Hough transform's bins are restricted to horizontal angles from -45 to 45 degrees, and from the peaks of its histogram of line parameters, find the corresponding pixels in the line image. When these detected lines end near the boundary of the sky region, we extend the line to the boundary. We also break up long line segments into smaller chunks to more accurately represent curved lines.

As illustrated in Figure 6-5 *right*, since we want to eliminate false points on extracted lines. For each pixel in the lines detected by our modified Hough transform, we create a vertical neighborhood (in our case six pixels above and below the line pixel). We then search for the peak contrast pixel in the vertical neighborhood to find the best corresponding point on the line.

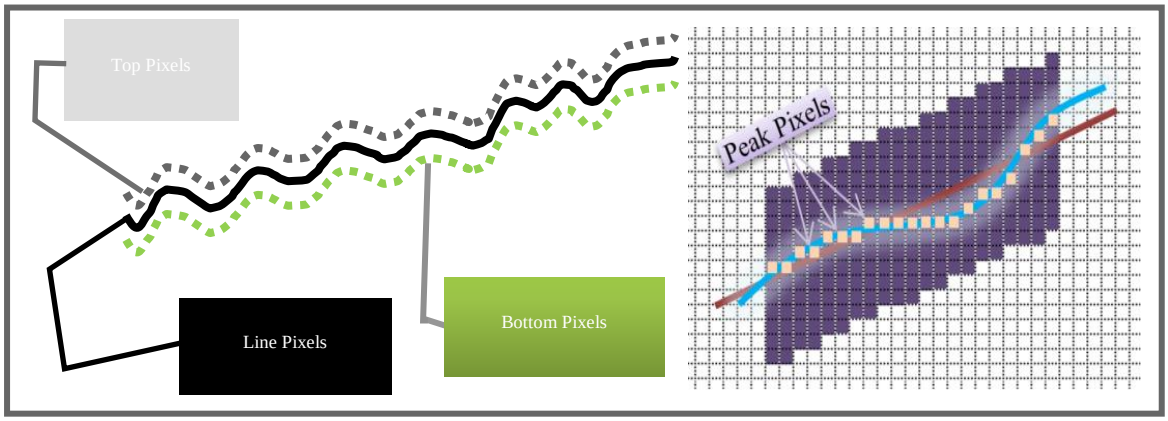


Figure 6-5 *(Left)* Choosing Top and Bottom Pixels of Partial Line
***(Right)* Finding peak pixels along a line segment**

For each neighborhood, we find the highest contrast pixel and fit a regression line to its neighboring pixels for each detected line segment. If the variance of the difference between these high-contrast pixels and the regression line exceeds a predefined threshold then we reject the line segment.

6.3 Linear Clutter Removal

In this step we pass a bilateral median filter over the image using neighborhood size (max_line_width*3, max_line_width*3), where max_line_width was defined in Section 3. Having found the peak pixels from the previous step, we create a new map consisting of peak pixels and their vertical neighbors within filter_width distance. filter_width refers to from the filter size which had the highest return for line detection. We replace each pixel in this new removal map with its median filter image value which was extracted at the beginning of the removal step (Figure 6-6).

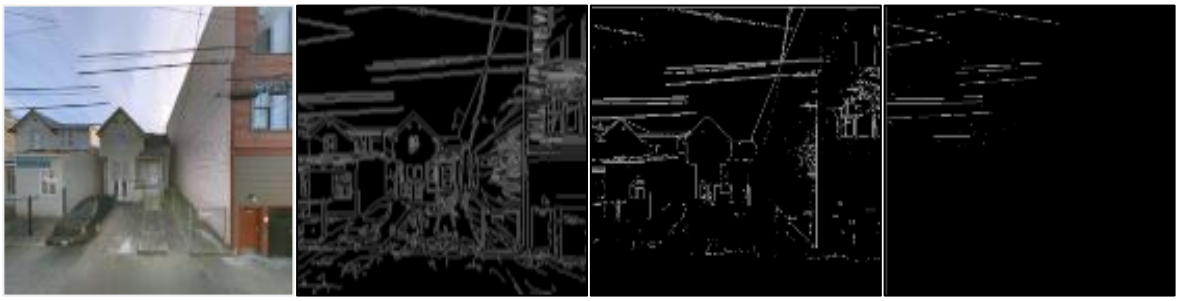
Figure 6-6 Blurring (Left) original image. (Right) blurred horizontal wires.

6.4 Experimental Results

We implemented the linear clutter detection and removal algorithm described in Sections 3 and 4 in MATLAB. We tested the performance of each component on 64 bit, 2.2GHZ computer. In our tests, we found that calculating the sky mask takes about 0.7 seconds in MATLAB. The subsequent wire detection steps require roughly 12 seconds of runtime per image (512x512 pixels). Blurring the image to remove linear clutter takes less than one tenth of one second. We predict that, if we

implement our method in C++ instead of MATLAB, a further performance improvement would be easily attainable. As mentioned earlier, the most important parameters that are needed for this algorithm are the min and max line width.

One of the most aspects of our method is that is that our unique filter which focuses only on extracting lines which belong to wires on high-contrast backgrounds. Our method avoids extracting edges and linear features on building facades. Figure 6-7 demonstrates the advantages of our method over two general edge detection techniques.



Original Image Canny Sobel Ours

Figure 6-7 Doubled lines in edge detection vs. single lines in our method

A challenge on our experiments was that the facades of the buildings which contained big sky colored regions (such as reflection of the sky on the windows) made the rejection fail on those regions and hence, blurred (Figure 6-8).

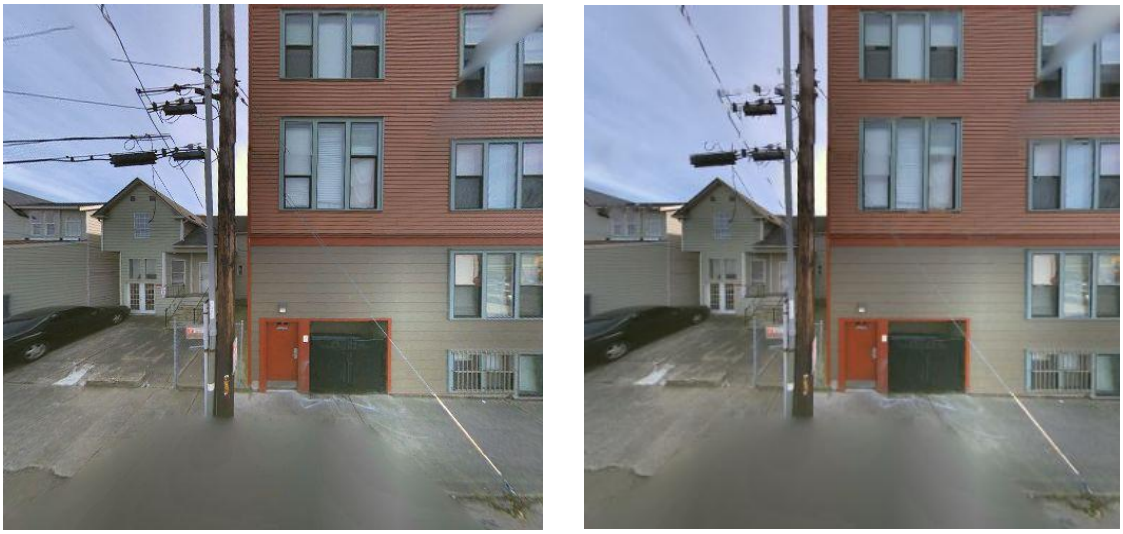


Figure 6-8 *(Left)* Original Image, *(Right)* linear clutter removal result. Problem is visible on blurred pixels on windows which match sky color and didn't get avoided.

Figure 6-9 Experimental Results on Different Urban Panoramas

Figure 6-9 shows some samples of real urban scene panoramas which their linear clutter has been removed using our technique. As it's visible from the images, the clutter in these panoramas has significantly been reduced.

Another fact to consider on our method is deciding how much blurring trees vs. removing all the visible clutter mattered. This affected how we chose the rejection threshold for gradient entropy. Figure 6-10 shows an example of choosing different entropy thresholds. Particularly on the left image, the evergreen tree top is blurred due to the low entropy rejection threshold.

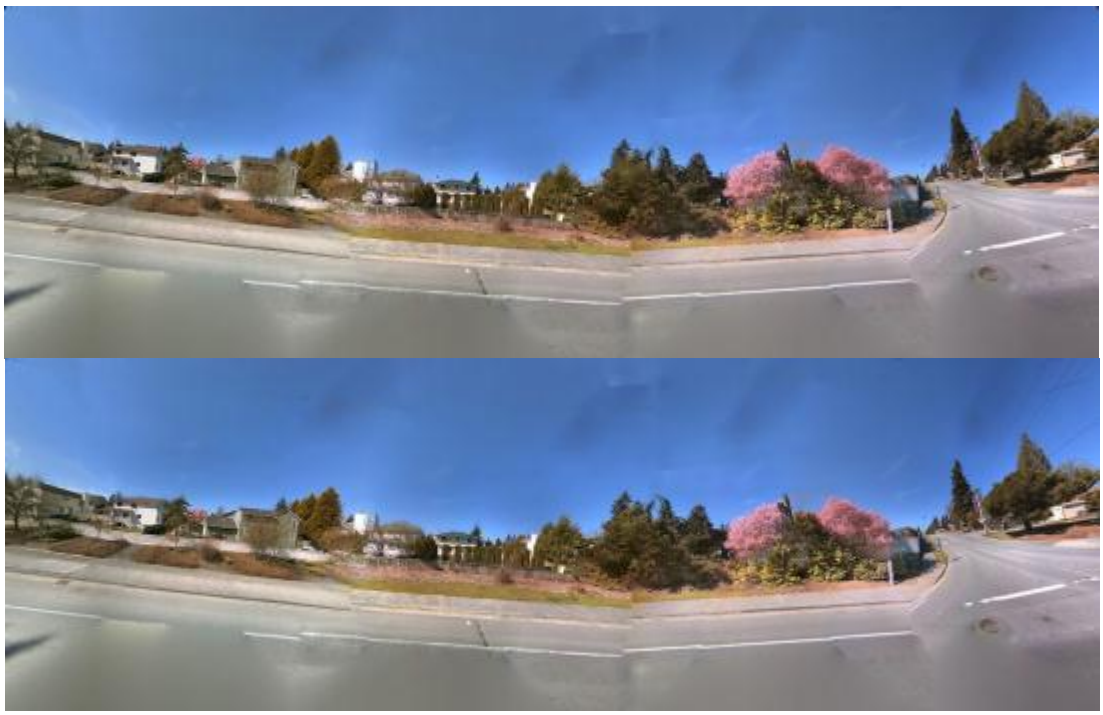


Figure 6-10 Effect of different rejection entropy thresholds on blurring the trees.
***(Left)* low threshold (look at the left big evergreen)**
***(Right)* high threshold**

6.5 Conclusion

We demonstrated a technique for identifying and removing line clutter from images. Our technique enhances panoramic scenes that contain power lines or other linear clutter. In future we could take a look at replacing the removed lines with clean Bezier curve replacements and synthetic telephone lines in order to create an exact match to the original scene. Our technique is already being integrated into a well-known urban navigation application.

Summary

In this dissertation, we offered novel advancements in acquired scene analysis and manipulation. Nowadays, user expectation from Virtual Environments has grown significantly, and as a result, research and exploration in this domain has gained a great attention. At the outset, we investigated different types of user interfaces for 3D Navigation and showed the importance of in depth case studies for their usage. In addition, we introduced three novel techniques for data manipulation.

We investigated importance of balance between complexity and efficiency of user interface in different environments. In particular, we conducted case studies on quite different applications to validate the generality of this issue. The applications varied from circuit design to 3D dance environment and GPS navigation. In our suggested dance composition software, the interface evolved from dancers' personal experience into an optimal combination of predetermined virtual scenes and real time control of the scene by a Wii haptic device.

It was discussed that based on the available bandwidth and processing power of a virtual environment, one can chose between 2D and 3D data. For example, when working with mobile devices, one cannot easily transmit huge 3D data. However, whenever high bandwidth and processing power is available, it is often better to use full 3D capabilities. In this thesis, we addressed both 2D and 3D data types in different applications and developed novel techniques to tackle with the challenges they caused. The ultimate result of our work provided robust enhancement of and detection in 2D and 3D data.

We studied two 2D applications which focused on robust facial feature detection and panoramic street image enhancement. The first application, which we named Meth Morph, analyzed 2D portrait images of human face and morphed them into a face as if the subject has been a drug addict. The developed technique required only a single mug shot image and performed the rest of the process in a fully automatic way. The robustness of the systems allowed us to give a live demo given to random visitors. The second application was on enhancing street panoramas where telephone wire clutter over the pace of stitched images seriously damaged the quality of street panoramas. We developed a fully automatic detection and removal of wire-like clutters in the images. Our results showed significant visual improvement over the created panoramas.

We also addressed issues in 3D data management and structure detection. The first challenge with manipulating 3D data is its huge size compared to 2D, which creates conflict with need to use them in real-time and interactive environments. We developed an efficient system based on a spatial data structure allowing real-time rendering and manipulation of a 3D point cloud. Most importantly, we developed a novel method for robust classification of curvilinear and surface like structures within a point cloud.

The mentioned components help remarkably improve the efficiency and quality of data processing, analysis and even transmission in the specified tasks.

Concluding Remarks

This thesis investigated some aspects in digital data acquisition, analysis and synthesis. The major conclusions resulted from this investigation listed and summarized as the following. Reference to a few examples from developed projects is provided in each case.

- Acquisition Resolution: Nowadays we have access to vast varieties of 2D and 3D digital data acquisition tools. An economical choice among these options must strike a balance between the amounts of details/precision to be delivered to the user versus available resources, such as bandwidth, computation power. For example, consider the task of estimating the 3D skeletal structure of a tree. Here, relying on 2D images as the input barely provides any meaningful representation of the 3D tree, due to the intricate structure of branches whose look significantly depends on the viewpoint. Moreover, in a map making company, there is ample access to storage and high tech acquisition tools such as LiDAR scanners. Hence, in that project we could choose LiDAR as our data collection tool. However, in another project, since the main goal was representing the extracted map making data to a mobile device with significantly low bandwidth, we had to either use stitched 2D images or develop more sophisticated algorithms to extract 3D models from data. Since extracting 3D models from street scans is a slow developing technology, we had to choose using only 2D stitched images along a street. Hence, our acquisition tool was only a simple 2D camera.
- Visualization Complexity: Given data acquired in the proper way, we studied additional constraints on visualization complexity caused by the renderer. For example, on a mobile phone, it would be very difficult to visualize information in full 3D view due to lack of

appropriate graphics technology on them. However, the same device is capable of easily displaying 2D information. As a result, in order to present different angles of an object to the user, the most feasible rendering is to project the appropriate angle onto a 2D image and display it to the user. At the other side of complexity spectrum lies 3D visualization for data such as point clouds. Here, if the 3D environment is not scanned at a uniform rate, the resulted point cloud can look very confusing to the user. However, this can be often resolved by adding intensity or, better if possible, color to the point cloud image. In addition, rendering each point as a big enough sphere is able to fill in the gaps between the samples and thus help it look more uniform. Finally, the most sophisticated solution to this problem is fitting simple geometric models to the data, which helps showing them as solid objects in the scene. However, fully automatic generation of models for this type of visualization is very challenging, and is still an open problem for research community. This thesis studied the issue it the specific setting of tree modeling.

- Data Analysis: Another challenging issue is making decision about proper way of detecting structures in 2D or 3D data. This task is specially challenging for 3D data. Notice that in 2D images, we work with dense and connected grid of pixels. This greatly facilitates the task of data analysis, as there well developed techniques for 2D image enhancement and segmentation. However, in a 3D setting, we often have to work with sparsely sampled data in a point cloud. Neighborhood connections are not included in the data and must be discovered by data analysis. Since the volume of 3D data is often huge, we have to use an appropriate data structure for processing the data such as KD-trees or Octrees. Local analysis of the data (i.e. within a small spatial neighborhood) can produce very unstable estimates about the structure of the data. In contrast, taking the global topology of the data into the account by any means, such as Markov random fields, can boost the quality of the estimates significantly. We thoroughly studied this problem and developed a tailored surface extraction technique for this purpose.

Related Publications

1. “Robust Classification of Curvilinear and Surface like Structures in 3D Point Cloud Data”, Mahsa Kamali, Matei Stroila, Jason Cho, Eric Schaffer, John C. Hart, Proceedings of ISVC 2011 as Springer Verlags Journal of Lecture Notes on Computer Science.
2. “MethMorph: Simulating Facial Deformation dueto Methamphatamine Usage”, Mahsa Kamali, Forrest Iandola, Hui Fang, John C. Hart, Proceedings of ISVC 2011 as Springer Verlags Journal of Lecture Notes on Computer Science.
3. “Linear Clutter Removal from Urban Panoramas”, Mahsa Kamali, Ido Omer, Forrest Iandola, Eyal Ofek, John C. Hart, Proceedings of ISVC 2011 as Springer Verlags Journal of Lecture Notes on Computer Science
4. “A High-Quality Low-Delay Remote Rendering System for 3D Video”, Shu Shi, Mahsa Kamali, Klara Nahrstedt, John C. Hart, Roy H. Campbell, In Proceedings of the 18th ACM International Conference on Multimedia (MM’10), pages 601-610, Firenze, Oct. 2010.
5. Patent Application: “Determining Travel Path Features Based on Retroreflectivity”, Inventors: Matei Nicolai Stroila, Xin Chen, Mahsa Kamali Moghaddam, Victor Lu, Bradley Dean Kohlmeyer, Publication date: 08/26/2010, Patent application number: 20100217529
6. “Advancing Interactive Collaborative Mediums through Tele-immersive Dance (TED): A Symbiotic Creativity and Design Environment for Art and Computer Science”, Renata Sheppard, Mahsa Kamali, Raoul Rivas, Morihiko Tamai, Zhenyu Yang, Wanmin Wu and Klara Nahrstedt, ACM International Conference on Multimedia (MM'08, full paper), Vancouver, BC, Canada, October 27–31, 2008.
7. “Tele-immersive Dance (TED): Evolution in Progress, ACM International Conference on Multimedia (MM'08, video program)”, Renata Sheppard, Mahsa Kamali, Morihiko Tamai, Raoul Rivas, Zhenyu Yang, Wanmin Wu and Klara Nahrstedt, Vancouver, BC, Canada, October 27–31, 2008.
8. "WiiView: A view control interface for 3D tele-immersive environments", Morihiko Tamai, Wanmin Wu, Renata M. Sheppard, Mahsa Kamali, Klara Nahrstedt, ICME, pp.1593-1594, 2008.

Bibliography

[1] "Wii at Nintendo." [Online]. Available: http://www.nintendo.com/wii. [Accessed: 25-Oct-2011].

[2] S. Halpern, "The iPad Revolution," *The New York Review of Books*, vol. 2010, no. 10(June 2010), pp. 1-6, 2010.

[3] M. R. Syamsuddin, C. H. Lee, and Y.-M. Kwon, "3D Virtual World and Haptic," *Screen*, pp. 3-4, 2009.

[4] R. Sheppard, M. Kamali, R. Rivas, Z. Yang, W. Wu, and K. Nahrstedt, "Tele-Immersive dance (TED): Evolution in Progress," *Computing*, pp. 1105-1106.

[5] M. Tamai, W. Wu, R. Sheppard, M. Kamali, and K. Nahrstedt, "WIIVIEW : A VIEW CONTROL INTERFACE FOR 3D TELE-IMMERSIVE ENVIRONMENTS Nara Institute of Science and Technology Graduate School of Information Science University of Illinois at Urbana-Champaign Department of Computer Science," *Options*, pp. 3-4, 2008.

[6] R. M. Sheppard, M. Kamali, R. Rivas, M. Tamai, W. Wu, and K. Nahrstedt, "Advancing Interactive Collaborative Mediums through Tele-immersive Dance (TED): A Symbiotic Creativity and Design Environment for Art and Computer Science," *Digital Media*, pp. 579-588, 2008.

[7] D. Siroker and P. Koomen, "Optimizely: A/B Testing." [Online]. Available: http://www.optimizely.com/whatisabtesting. [Accessed: 07-Nov-2011].

[8] "2D vs 3D Map View...Also what zoom level do U like? - Magellan Maestro Forum." [Online]. Available: http://forums.gpsreview.net/viewtopic.php?t=9211. [Accessed: 07-Nov-2011].

[9] "3D Laser Mapping." [Online]. Available: http://www.3dlasermapping.com/index.php?option=com_content&view=article&id=20&Itemid=54. [Accessed: 07-Nov-2011].

[10] S. Shi, M. Kamali, K. Nahrstedt, J. C. Hart, and R. H. Campbell, "A High-Quality Low-Delay Remote Rendering System for 3D Video," *System*, pp. 601-610.

[11] K. Engel, R. Westermann, and T. Ertl, "Isosurface extraction techniques for Web-based volume visualization," *Proceedings Visualization '99 (Cat. No.99CB37067)*, vol. Vi, pp. 139-519, 1999.

[12] H. Yokoyama, "Efficient noise reduction of laser scanning data for archaeological survey," *Archives*, vol. XXXIV, pp. 241-244, 2001.

[13] Tom Lane and T. Lane, *JPEG image compression FAQ*. 1999.

[14] S. Swarup, T. Oczer, S. R. Ray, and T. J. Anastasio, "A self-airing camera based on neurophysical principles," *Proceedings of the International Joint Conference on Neural Networks, 2003.*, vol. 4, pp. 3201-3206.

[15] I. T. Jolliffe, *Principal Component Analysis*, Springer S. NY: Springer, 2002, p. 487.

[16] X. Chen, W. R. Street, and R. Wang, "Next Generation Map Making : Geo-Referenced Ground-Level LIDAR Point Clouds for Automatic Retro-Reflective Road Feature Extraction," *Image Processing*, no. 3.

[17] L. Zebedin, J. Bauer, K. Karner, and H. Bischof, "Fusion of Feature- and Area-Based Information for Urban Buildings Modeling from Aerial Imagery," *Image (Rochester, N.Y.)*, no. 813397, pp. 873-886, 2008.

[18] C. Xin et al., "Towards Next Generation Map Making," in *IEEE International Conference on Multimedia & Expo*, 2009, vol. 0, pp. 1676-1683.

[19] W. Shen, "Building boundary extraction based on lidar point clouds data," in *ISPRS08*, 2008, vol. XXXVII, p. 157.

[20] S. Belongie and J. Malik, "Matching with shape contexts," in *Proceedings IEEE Workshop on Content-Based Access of Image and Video Libraries*, pp. 20-26.

[21] T. Rabbani and F. Van Den Heuvel, "EFFICIENT HOUGH TRANSFORM FOR AUTOMATIC DETECTION OF CYLINDERS IN POINT CLOUDS," *Transform*, vol. 3, pp. 60–65, 2005.

[22] S. Paris and F. Durand, "A Fast Approximation of the Bilateral Filter Using a Signal Processing Approach," *International Journal of Computer Vision*, vol. 81, no. 1, pp. 24-52, 2007.

[23] Z. Zhang, A. Ganesh, X. Liang, and Y. Ma, "TILT: Transform Invariant Low-rank Textures," *International Journal of Computer Vision*, pp. 1-14, 2010.

[24] M. Kamali, E. Ofek, F. Iandola, I. Omer, and J. C. Hart, "Linear Clutte r Removal from Urban Panoramas," *Lecture Notes in Computer Science*, vol. 6939, no. Advances in Visual Computing, pp. 85-94, 2011.

[25] K. Karantzalos and N. Paragios, "Large-Scale Building Reconstruction Through Information Fusion and 3-D Priors," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 48, no. 5, pp. 2283-2296, May. 2010.

[26] M. Kamali and F. Iandola, "MethMorph: simulating facial deformation due to methamphatamine usage," *Advances in Visual Computing*, 2011.

[27] N. Nicosia, R. L. Pacula, B. Kilmer, R. Lundberg, and J. Chiesa, "The Economic Cost of Methamphetamine Use in the United States, 2005." RAND Corporation, 2009.

[28] Thomas M. Siebel and Steven A. Mange, “The Montana meth project: ‘unselling’ a dangerous drug,” *Stanford Law and Policy Review* , vol. 20, pp. 405–416, 2009.

[29] “The Meth Project.” [Online]. Available: http://notevenonce.com/. [Accessed: 24-Nov-2011].

[30] P. Viola and M. J. Jones, “Robust Real-Time Face Detection,” *International Journal of Computer Vision*, vol. 57, no. 2, pp. 137-154, May. 2004.

[31] M. J. Paul Viola, “Robust Real-time Object Detection,” *International Conference on Computer Vision*, p. 747, 2001.

[32] M. Castrillón and O. Déniz, “A comparison of face and facial feature detectors based on the Viola–Jones general object detection framework,” *Machine Vision and …*, 2011.

[33] M. Santana and O. Déniz-Suárez, “Face and facial feature detection evaluation-performance evaluation of public domain haar detectors for face and facial feature detection,” *VISAPP (2)*, 2008.

[34] “FaceDetection - OpenCV Wiki.” [Online]. Available: http://opencv.willowgarage.com/wiki/FaceDetection.

[35] R. Lienhart, “Empirical analysis of detection cascades of boosted classifiers for rapid object detection,” *Pattern Recognition*, 2003.

[36] Y. Li and J. Sun, “Lazy snapping,” *ACM Transactions on Graphics (ToG)*, 2004.

[37] Y. Bediz and G. B. Akar, “View point tracking for 3d display systems,” *3rd European Signal Processing Conference*, 2005.

[38] J. Canny, “A Computational Approach to Edge Detection,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 8, no. 6, pp. 679-698, Nov. 1986.

[39] M. CASTRILLON, O. DENIZ, C. GUERRA, and M. HERNANDEZ, “ENCARA2: Real-time detection of multiple faces at different resolutions in video streams,” *Journal of Visual Communication and Image Representation*, vol. 18, no. 2, pp. 130-140, Apr. 2007.

[40] M. Kass, “Snakes: Active contour models,” *International journal of computer …*, 1988.

[41] M. K. Moghaddam and R. Safabakhsh, “TASOM-based Lip Tracking Using the Color and Geometry of the Face,” in *Fourth International Conference on Machine Learning and Applications (ICMLA’05)*, pp. 63-68.

[42] H. Fang and J. C. Hart, “Detail preserving shape deformation in image editing,” *ACM Transactions on Graphics*, vol. 26, no. 3, p. 12, Jul. 2007.

[43] P. Pérez, M. Gangnet, and A. Blake, “Poisson image editing,” *ACM Transactions on Graphics*, vol. 22, no. 3, p. 313, Jul. 2003.

[44] Jian Sun and Lu Yuan, "Image completion with structure propagation," *ACM Transactions on Graphics*, p. 861--868, 2005.

[45] J. C. H. Hui Fang John, "Textureshop: Texture Synthesis as a Photograph Editing Tool," *ACM Transactions on Graphics*, vol. 23, no. 3, pp. 354-359, 2004.

[46] Hui Fang and John C. Hart, "Methods and systems for image ... - Google Patents," .

[47] R. Unnikrishnan, J.-F. Lalonde, N. Vandapel, and M. Hebert, "Scale Selection for the Analysis of Point-Sampled Curves," *Third International Symposium on 3D Data Processing Visualization and Transmission 3DPVT06*, pp. 1026-1033, 2006.

[48] A. Golovinskiy, V. G. Kim, and T. Funkhouser, "Shape-based recognition of 3D point clouds in urban environments," *2009 IEEE 12th International Conference on Computer Vision*, no. Iccv, pp. 2154-2161, 2009.

[49] Y. Boykov and V. Kolmogorov, "An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision.," *IEEE transactions on pattern analysis and machine intelligence*, vol. 26, no. 9, pp. 1124-37, Sep. 2004.

[50] D. Munoz, N. Vandapel, and M. Hebert, "Directional Associative Markov Network for 3-D Point Cloud Classification," *Network*, 2008.

[51] D. Munoz, N. Vandapel, and M. Hebert, "Onboard Contextual Classification of 3-D Point Clouds with Learned High-order Markov Random Fields," in *Robotics Institute*, 2009.

[52] I. T. Jolliffe, *Principal Component Analysis*, 2nd ed. Springer Series in Statistics, 2002.

[53] J. Lalonde, R. Unnikrishnan, N. Vandapel, and M. Hebert, "Scale Selection for Classification of Point-Sampled 3-D Surfaces," in *Fifth International Conference on 3-D Digital Imaging and Modeling (3DIM'05)*, 2005, pp. 285-292.

[54] X. Xiong, D. Munoz, J. A. Bagnell, and M. Hebert, "3-D Scene Analysis via Sequenced Predictions over Points and Regions," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2011.

[55] D. Munoz, J. A. Bagnell, N. Vandapel, and M. Hebert, "Contextual Classification with Functional Max-Margin Markov Networks," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2009.

[56] R. Shapovalov, A. Velizhev, and O. Barinova, "NON-ASSOCIATIVE MARKOV NETWORKS FOR 3D POINT CLOUD CLASSIFICATION," *Photogrammetric Computer Vision and Image Analysis*, vol. XXXVIII, no. 3, pp. 103-108, 2010.

[57] D. P. Bertsekas, A. Nedić, and A. E. Ozdaglar, *Convex analysis and optimization*. Athena Scientific, 2003.

[58] J. Kopf, B. Chen, R. Szeliski, and M. Cohen, "Street Slide: Browsing Street Level Imagery," *ACM Transactions on Graphics (Proceedings of SIGGRAPH 2010)*, vol. 29, no. 4, p. 96:1 -- 96:8, 2010.

[59] A. Agarwala, M. Agrawala, M. Cohen, D. Salesin, and R. Szeliski, “Photographing long scenes with multi-viewpoint panoramas,” *ACM Transactions on Graphics*, vol. 25, no. 3, p. 853, Jul. 2006.

[60] K. Pulli, M. Tico, Y. Xiong, P. M. Road, and P. Alto, “Mobile Panoramic Imaging System,” *ECVW*, 2010.

[61] A. Rav-Acha, G. Engel, and S. Peleg, “Minimal Aspect Distortion (MAD) Mosaicing of Long Scenes,” *International Journal of Computer Vision*, vol. 78, no. 2-3, pp. 187-206, Nov. 2007.

[62] A. Agarwala, M. Agrawala, M. Cohen, D. Salesin, and R. Szeliski, “Photographing long scenes with multi-viewpoint panoramas,” *ACM Transactions on Graphics*, vol. 25, no. 3, p. 853, Jul. 2006.

[63] S. Vallance and P. Calder, “Multi-perspective images for visualisation,” in *Pan-Sydney area workshop on Visual information processing*, 2001, pp. 69-76.

[64] A. Roman, G. Garg, and M. Levoy, “Interactive design of multi-perspective images for visualizing urban landscapes,” in *IEEE Visualization 2004*, 2004, pp. 537-544.

[65] A. Roman and H. P. A. Lensch, “Eurographics Symposium on Rendering (2006) Tomas Akenine-Muller and Wolfgang Heidrich (Editors) Automatic Multiperspective Images.” .

[66] R. Szeliski, “Image Alignment and Stitching: A Tutorial,” *Foundations and Trends® in Computer Graphics and Vision*, vol. 2, no. 1, pp. 1-104, 2006.

[67] G. Yan, C. Li, G. Zhou, W. Zhang, and X. Li, “Automatic Extraction of Power Lines From Aerial Images,” *IEEE Geoscience and Remote Sensing Letters*, vol. 4, no. 3, pp. 387-391, Jul. 2007.

[68] C. Mu, J. Yu, Y. Feng, and J. Cai, “Power lines extraction from aerial images based on Gabor filter,” in *Proceedings of SPIE*, 2009, vol. 7492, p. 74923P-74923P-8.

[69] A. N. Hirani and T. Totsuka, “Combining frequency and spatial domain information for fast interactive image noise removal,” *Proceedings of the 23rd annual conference on Computer graphics and interactive techniques - SIGGRAPH ’96*, pp. 269-276, 1996.

[70] A. Hirani and T. Totsuka, “Projection Based Method for Scratch and Wire Removal from Digital Images.” United State Patents, p. 8, 1996.

[71] G. Beylkin, “Discrete Radon Transform,” no. 2, pp. 162-172, 1987.

[72] M. V. Ginkel, C. L. L. Hendriks, and L. J. V. Vliet, “A short introduction to the Radon and Hough transforms and how they relate to each other,” *Reading*, 2004.

[73] S. Battiato, A. Capra, S. Curti, and M. L. Cascia, “3D Stereoscopic Image Pairs by Depth-Map Generation,” in *2nd International Symposium on 3D Data Processing, Visualization, and Transmission*, 2004, pp. 124-131.

[74] L. Tao, L. Yuan, and J. Sun, “SkyFinder,” *ACM Transactions on Graphics*, vol. 28, no. 3, p. 1, Jul. 2009.

[75] D. Hoiem, A. A. Efros, and M. Hebert, “Closing the loop in scene interpretation,” in *2008 IEEE Conference on Computer Vision and Pattern Recognition*, 2008, pp. 1-8.

[76] H. Rheingold, *Tools for Thought: The History and Future of Mind-Expanding Technology*. Cambridge, MA: MIT Press, 2000.

Appendix A

Touch Screen Maps Interface Design Enhancement

A.1 Basic Rules (Based on Win8 design principles):

- Buttons should be at least 10mm wide and 4mm apart.
- Be forgiving (accidental touch actions should be reversible)
- Require Additional Feedback for destructive actions.
- Design with fat finger problem in mind.

A.2 Standard actions are (based on Win8 touch design principles):

- Panning
- Zooming
- Two Finger Rotation
- Avoid time based gestures (such as press and hold)
- Avoid creating additional gestures (such as drawing P for print)

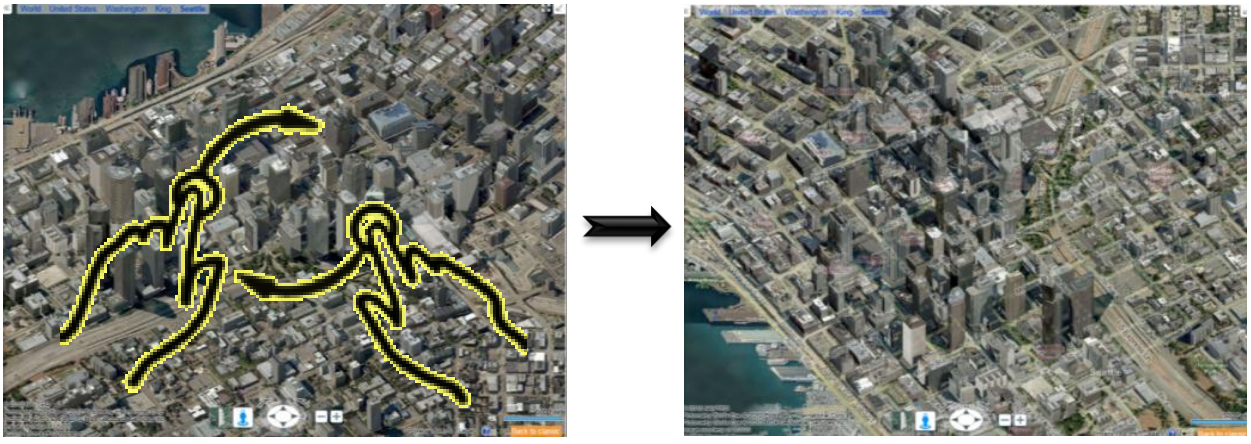


Figure A-1 Map Rotations using two finger rotations gesture

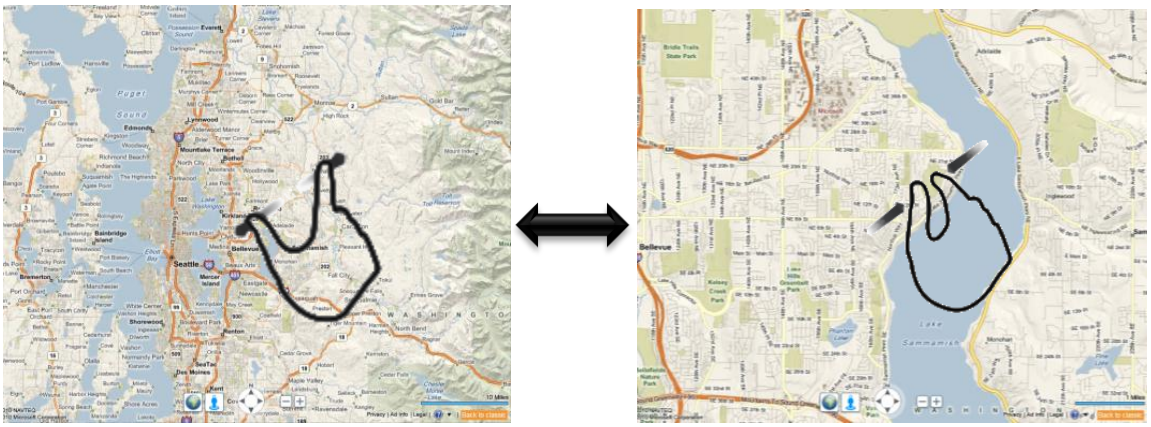


Figure A-2 All Map Modes Zoom in/out using pinch gesture

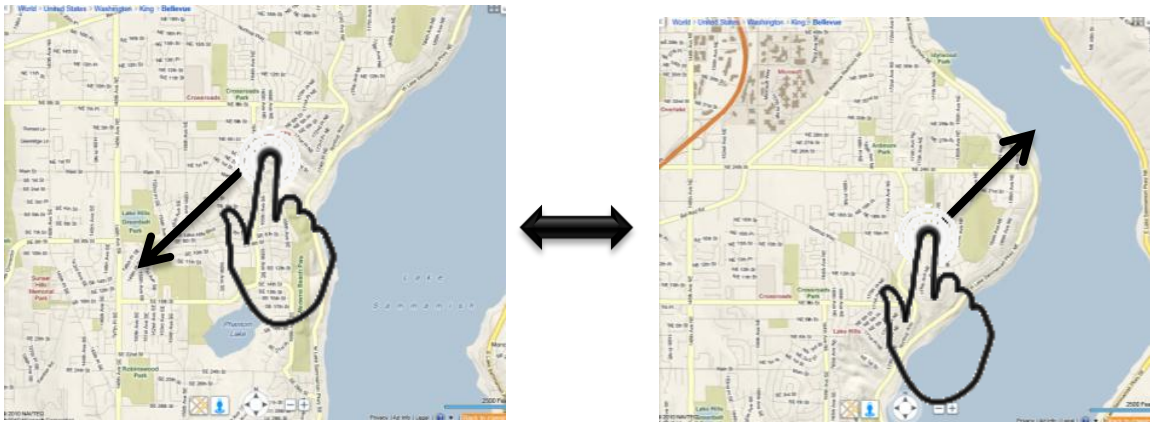


Figure A-3 All Map Modes Panning

Figure A-4 Single tap move forward on Human Scale (not in the scope of this spec)

Figure A-5 Zooming in/out with pinching gesture on Human Scale (not in the scope of this spec)

A.3 Performance

- Touch events shouldn't cause any key performance metrics to fall below release criteria.
- Need to make sure we are going to use the touch API provided from IE9~10 to support parallel threads for touch events for optimal performance.

A.4 Release Criteria / Success Metrics

List the release criteria & other success metrics that are relevant to this feature. A subset of the metrics should make up the Release Criteria. These are the critical metrics that will be used to make a go/no-go decision during flights. If it is a Release Criteria, mark "Y" in the RC column. All metrics should be measurable. Work with the instrumentation team on a measurement plan and include it.

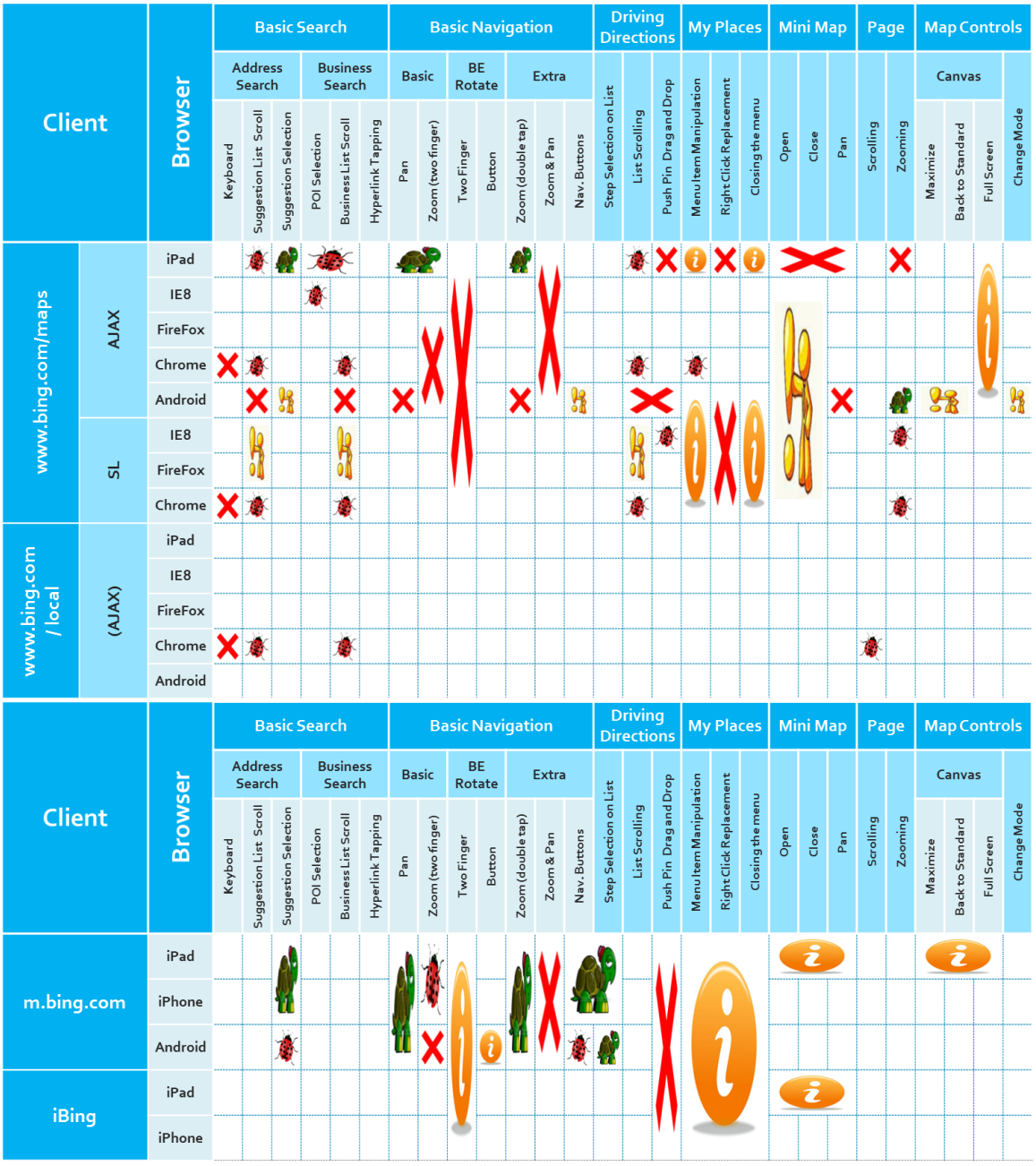
Success Metric	Target	Notes
No Fat Finger Issue	Buttons, Hyperlinks, Menus, Check boxes, Radio Buttons, Drop Down Lists	Count the percentage of items that currently have this issue.
All Actions Must work without need for Hover	Hover Actions	If there is a hover action, it should be achievable without using it as well. (ex. POI Menus appear both by hoover and single click/tap)
No Touch Actions should cause flicker	POI menus, Right Click Menu, Drag and Drop,	Tapping must act consistently.
User Experience	User Study	Conduct a user study and see how satisfied they are with current touch implementation

Below are the issues discovered in iPad navigation software from Bing Maps grouped by platform.

Legend					
	Works Fine ☺		Not Supported		Has a Bug
Too Slow!		Doesn't Work ☹		Fat Finger Trouble	
					Not Applicable

Client		Browser	Basic Search						Basic Navigation								Driving Directions			My Places			Mini Map			Page		Map Controls				
			Address Search			Business Search			Basic		BE Rotate		Extra															Canvas				
			Keyboard	Suggestion List Scroll	Suggestion Selection	POI Selection	Business List Scroll	Hyperlink Tapping	Pan	Zoom (two finger)	Two Finger	Button	Zoom (double tap)	Zoom & Pan	Nav. Buttons	Step Selection on List	List Scrolling	Push Pin Drag and Drop	Menu Item Manipulation	Right Click Replacement	Closing the menu	Open	Close	Pan	Scrolling	Zooming	Maximize	Back to Standard	Full Screen	Change Mode
www.bing.com/maps	AJAX	iPad	✓	🐞	🐢	🐞	🐞	✓	🐢	🐢	✗	✓	🐢	✗	✓	✓	🐞	✗	ⓘ	✗	ⓘ	✗	✗	✗	⊘	✗	✓	✓	ⓘ	✓
		IE8	✓	✓	✓	🐞	✓	✓	✓	✓	✗	✓	✓	✗	✓	✓	✓	✓	✓	✓	✓	🤦	🤦	✓	⊘	✓	✓	✓	ⓘ	✓
		FireFox	✓	✓	✓	✓	✓	✓	✓	✗	✗	✓	✓	✗	✓	✓	✓	✓	✓	✓	✓	🤦	🤦	✓	⊘	✓	✓	✓	ⓘ	✓
		Chrome	✗	🐞	✓	✓	🐞	✓	✓	✗	✗	✓	✓	✗	✓	✓	🐞	✓	🐞	✓	✓	🤦	🤦	✓	⊘	✓	✓	✓	ⓘ	✓
		Android	✓	✗	🤦	✓	✗	✓	✗	✗	✗	✓	✗	✗	🤦	✓	✗	✗	ⓘ	✗	ⓘ	🤦	🤦	✗	✓	🐢	🤦	🤦	ⓘ	🤦
	SL	IE8	✓	🤦	✓	✓	🤦	✓	✓	✓	✗	✓	✓	✓	✓	✓	🤦	🐞	ⓘ	✗	ⓘ	🤦	🤦	✓	⊘	🐞	✓	✓	✓	✓
		FireFox	✓	🤦	✓	✓	🤦	✓	✓	✓	✗	✓	✓	✓	✓	✓	🤦	✓	ⓘ	✗	ⓘ	🤦	🤦	✓	⊘	✓	✓	✓	✓	✓
		Chrome	✗	🐞	✓	✓	🐞	✓	✓	✓	✗	✓	✓	✓	✓	✓	🐞	✓	ⓘ	✗	ⓘ	🤦	🤦	✓	⊘	🐞	✓	✓	✓	✓
www.bing.com / local	(AJAX)	iPad	✓	✓	✓	⊘	✓	✓	⊘	⊘	⊘	⊘	⊘	⊘	⊘	⊘	⊘	⊘	⊘	⊘	⊘	⊘	⊘	⊘	✓	✓	⊘	⊘	⊘	⊘
		IE8	✓	✓	✓	⊘	✓	✓	⊘	⊘	⊘	⊘	⊘	⊘	⊘	⊘	⊘	⊘	⊘	⊘	⊘	⊘	⊘	⊘	✓	✓	⊘	⊘	⊘	⊘
		FireFox	✓	✓	✓	⊘	✓	✓	⊘	⊘	⊘	⊘	⊘	⊘	⊘	⊘	⊘	⊘	⊘	⊘	⊘	⊘	⊘	⊘	✓	✓	⊘	⊘	⊘	⊘
		Chrome	✗	🐞	✓	⊘	🐞	✓	⊘	⊘	⊘	⊘	⊘	⊘	⊘	⊘	⊘	⊘	⊘	⊘	⊘	⊘	⊘	⊘	🐞	✓	⊘	⊘	⊘	⊘
		Android	✓	✓	✓	⊘	✓	✓	⊘	⊘	⊘	⊘	⊘	⊘	⊘	⊘	⊘	⊘	⊘	⊘	⊘	⊘	⊘	⊘	✓	✓	⊘	⊘	⊘	⊘

Client	Browser	Basic Search						Basic Navigation								Driving Directions			My Places			Mini Map			Page		Map Controls				
		Address Search			Business Search			Basic		BE Rotate		Extra															Canvas				
		Keyboard	Suggestion List Scroll	Suggestion Selection	POI Selection	Business List Scroll	Hyperlink Tapping	Pan	Zoom (two finger)	Two Finger	Button	Zoom (double tap)	Zoom & Pan	Nav. Buttons	Step Selection on List	List Scrolling	Push Pin Drag and Drop	Menu Item Manipulation	Right Click Replacement	Closing the menu	Open	Close	Pan	Scrolling	Zooming	Maximize	Back to Standard	Full Screen	Change Mode
m.bing.com	iPad	✓	✓	🐢	✓	✓	✓	🐢	🐞	ⓘ	✓	🐢	✗	🐢	🐢	✓	✗	ⓘ	ⓘ	ⓘ	ⓘ	ⓘ	ⓘ	✓	⊘	⊘	⊘	⊘	✓
	iPhone	✓	✓	🐢	✓	✓	✓	🐢	🐞	ⓘ	✓	🐢	✗	🐢	🐢	✓	✗	ⓘ	ⓘ	ⓘ	⊘	⊘	⊘	⊘	⊘	⊘	⊘	⊘	✓
	Android	✓	✓	🐞	✓	✓	✓	🐢	✗	ⓘ	ⓘ	🐢	✗	🐢	🐞	✓	✗	ⓘ	ⓘ	ⓘ	⊘	⊘	⊘	⊘	⊘	⊘	⊘	⊘	🐞
iBing	iPad	✓	✓	✓	✓	✓	✓	✓	✓	ⓘ	✓	✓	✓	✓	✓	✓	✗	ⓘ	ⓘ	ⓘ	ⓘ	ⓘ	ⓘ	✓	⊘	⊘	⊘	✓	✓
	iPhone	✓	✓	✓	✓	✓	✓	✓	✓	ⓘ	✓	✓	✓	✓	✓	✓	✗	ⓘ	ⓘ	ⓘ	⊘	⊘	⊘	⊘	⊘	⊘	⊘	⊘	✓


Client
Browser
Basic Search
Address Search
Business Search
Keyboard
Suggestion List Scroll
Suggestion Selection
POI Selection
Business List Scroll
Hyperlink Tapping
Basic Navigation
Basic
BE Rotate
Extra
Pan
Zoom (two finger)
Two Finger
Button
Zoom (double tap)
Zoom & Pan
Nav. Buttons
Driving Directions
Step Selection on List
List Scrolling
Push Pin Drag and Drop
My Places
Menu Item Manipulation
Right Click Replacement
Closing the menu
Mini Map
Open
Close
Pan
Page
Scrolling
Zooming
Map Controls
Canvas
Maximize
Back to Standard
Full Screen
Change Mode
www.bing.com/maps
AJAX
SL
www.bing.com / local
(AJAX)
iPad
IE8
FireFox
Chrome
Android
iPad
IE8
FireFox
Chrome
Android
Client
Browser
Basic Search
Address Search
Business Search
Keyboard
Suggestion List Scroll
Suggestion Selection
POI Selection
Business List Scroll
Hyperlink Tapping
Basic Navigation
Basic
BE Rotate
Extra
Pan
Zoom (two finger)
Two Finger
Button
Zoom (double tap)
Zoom & Pan
Nav. Buttons
Driving Directions
Step Selection on List
List Scrolling
Push Pin Drag and Drop
My Places
Menu Item Manipulation
Right Click Replacement
Closing the menu
Mini Map
Open
Close
Pan
Page
Scrolling
Zooming
Map Controls
Canvas
Maximize
Back to Standard
Full Screen
Change Mode
m.bing.com
iPad
iPhone
Android
iBing
iPad
iPhone